DISS. ETH NO. 25498

UNSUPERVISED LEARNING: MODEL-BASED CLUSTERING AND LEARNED COMPRESSION

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH
(Dr. sc. ETH Zurich)

presented by

MICHAEL TOBIAS TSCHANNEN

MSc ETH in Electrical Engineering and Information Technology

born on 03.08.1988
citizen of Zürich ZH

accepted on the recommendation of

Prof. Dr. Helmut Bölcskei,	examiner
Prof. Dr. Andreas Krause,	co-examiner

2018

Abstract

This thesis addresses two central tasks prevalent in many modern data processing, storage, and transmission pipelines: Clustering and compression. Specifically, in the first and the second part of this thesis, we study the problems of subspace clustering and random process clustering, respectively. While clustering problems are arguably among the most archetypal problems in unsupervised learning, compression methods are traditionally hand designed. In the third and fourth part of this thesis, we leverage machine learning techniques for compression, a trend that only emerged recently. In more detail, we propose a deep generative model-based framework for lossy data compression on one hand, and we study compression of neural network models for inference on resource-constrained hardware on the other hand.

Subspace clustering, the focus of the first part of this thesis, refers to the problem of clustering unlabeled high-dimensional data points into a union of unknown low-dimensional subspaces. Out of the plethora of subspace clustering algorithms, the sparse subspace clustering (SSC) algorithm (Elhamifar and Vidal, 2013) has attracted significant attention thanks to excellent clustering performance in practical applications. SSC performs spectral clustering based on an adjacency matrix obtained by sparsely representing each data point in terms of all the other data points via the Lasso. When the number of data points is large or the dimension of the ambient space is high, the computational complexity of SSC quickly becomes prohibitive. In this case, replacing the Lasso by the greedy orthogonal matching pursuit (OMP) algorithm results in significantly lower computational complexity, while often yielding comparable performance (Dyer et al., 2013). The main contribution of the first part of the thesis is an analytical performance characterization of the resulting SSC-OMP algorithm for noisy data. Moreover, we introduce and analyze the SSC-matching pursuit (SSC-MP) algorithm, which employs MP in lieu of OMP. Both SSC-OMP and SSC-MP are proven to succeed even when the subspaces underlying the data intersect and when the data points are contaminated by severe noise. Our experiments show that SSC-MP compares very favorably to other sparsity-based subspace clustering algorithms, both in terms of clustering performance and running time. In addition, we find that, in contrast to SSC-OMP, the performance of SSC-MP is very robust with respect to the choice of parameters in the stopping criteria.

The second part of this thesis deals with the problem of clustering noisy finite-length observations of stationary ergodic random processes according to their generative models without prior knowledge of the model statistics and the number of generative models. Two algorithms, both using the L^1-distance between estimated power spectral densities (PSDs) as a measure of dissimilarity, are analyzed. The first one, termed nearest neighbor process clustering (NNPC), relies on partitioning the nearest neighbor graph of the observations via spectral clustering. The second algorithm consists of a single k-means iteration with farthest point initialization and was considered before in the literature, albeit with a different dissimilarity measure and with asymptotic performance results only. We prove that both algorithms succeed with high probability in the presence of noise and missing entries, and even when the generative process PSDs overlap significantly, all provided that the observation length is sufficiently large. Our results quantify the tradeoff between the overlap of the generative process PSDs, the observation length, the fraction of missing entries, and the noise variance. Furthermore, we provide extensive numerical results for synthetic and real data and find that NNPC outperforms state-of-the-art algorithms in human motion sequence clustering.

In the third part of this thesis, we propose and study the problem of distribution-preserving lossy compression. Motivated by recent advances in extreme image compression which allow to maintain artifact-free reconstructions even at very low bitrates, we propose to optimize the rate-distortion tradeoff under the constraint that the reconstructed samples follow the distribution of the training data. Such a compression system recovers both ends of the spectrum: On one hand, at zero bitrate it learns a generative model of the data, and at high enough bitrates it achieves perfect reconstruction. Furthermore, for intermediate bitrates it smoothly interpolates between learning a generative model of the training data and perfectly reconstructing the training samples. We study several methods to approximately solve the proposed optimization problem, including a novel combination of Wasserstein generative adversarial networks and Wasserstein autoencoders, and present an extensive theoretical and empirical characterization of the proposed compression systems.

The fourth and last part of this thesis targets hardware-friendly compression of neural network models in the sense that the compressed models require only few multiplications at inference time. Specifically, we perform end-to-end learning of low-cost approximations of (generalized) matrix multiplications in deep neural network (DNN) layers by casting matrix multiplications as 2-layer sum-product networks (SPNs) (arithmetic circuits) and learning their (ternary) edge weights from data. The SPNs disentangle multiplication and addition operations and enable us to impose a budget on the number of multiplication operations. Combining our method with knowledge distillation techniques and applying it to image classification and language modeling DNNs, we obtain a first-of-a-kind reduction in number of multiplications (over 99.5%) while maintaining the predictive performance of the full-precision models. Finally, we demonstrate that the proposed framework is able to rediscover Strassen's matrix multiplication algorithm, learning to multiply 2×2 matrices using only 7 multiplications instead of 8.

Kurzfassung

Diese Dissertation behandelt zwei Datenverarbeitungsprobleme, die zentraler Bestandteil vieler moderner Datenverarbeitungs-, Speicher- und Übertragungssysteme sind: Clusteranalyse und Komprimierung. Im ersten und zweiten Teil dieser Dissertation betrachten wir das Clustern von Datenpunkten in Unterräume (englisch: subspace clustering) sowie das Clustern von Zufallsprozessen. Während Clusteranalyse wohl zu den ältesten Problemen im Bereich des unbeaufsichtigten Lernens gehört, werden Kompressionsverfahren traditionell von Hand entwickelt. Im dritten und vierten Teil dieser Dissertation setzen wir maschinelle Lerntechniken ein um Kompressionsalgorithmen zu entwickeln—ein Trend, der erst kürzlich aufkam. Wir schlagen einerseits ein System für die verlustbehaftete Datenkomprimierung basierend auf generativen Modellen vor und behandeln andererseits die Komprimierung neuronaler Netzwerke für deren Einsatz auf Hardware mit eingeschränkten Ressourcen.

Subspace clustering, der Fokus des ersten Teils dieser Dissertation, bezieht sich auf das Problem, hochdimensionale Datenpunkte in eine Vereinigung von unbekannten niedrigdimensionalen Unterräumen zu clustern. Aus der Vielzahl von bekannten subspace clustering Algorithmen hat der sparse subspace clustering (SSC) Algorithmus (Elhamifar und Vidal, 2013) aufgrund seiner exzellenten Clustering-Genauigkeit in praktischen Anwendungen erhebliche Aufmerksamkeit auf sich gezogen. SSC wendet spektrales Clustern auf eine Adjazenzmatrix an, die aus der spärlichen linearen Darstellung jedes Datenpunkts durch alle anderen Datenpunkte, berechnet mittels Lasso, aufgebaut wird.

Wenn die Anzahl der Datenpunkte gross ist oder die Dimension des Umgebungsraums hoch, wird die Rechenkomplexität von SSC schnell prohibitiv. In diesem Fall führt die Verwendung des orthogonal matching pursuit (OMP) Algorithmus anstelle von Lasso zu einer signifikant geringeren Rechenkomplexität, während die Clustering-Genauigkeit oft unbeinträchtigt bleibt (Dyer et al., 2013). Der Hauptbeitrag des ersten Teils der Dissertation ist eine statistische Analyse des resultierenden SSC-OMP Algorithmus für verrauschte Daten. Darüber hinaus führen wir den SSC-matching pursuit (SSC-MP) Algorithmus ein, der MP anstelle von OMP verwendet. Wir beweisen, dass SSC-OMP und SSC-MP erfolgreich clustern können wenn die den Daten zugrunde liegenden Unterräume überlappen und wenn der Rauschpegel hoch ist. Unsere Experimente zeigen, dass SSC-MP kompetitiv ist mit anderen subspace clustering Algorithmen, sowohl hinsichtlich der Clustering-Genauigkeit als auch der Laufzeit. Ausserdem beobachten wir, dass die Clustering-Genauigkeit von SSC-MP, im Gegensatz zu SSC-OMP, sehr robust gegenüber der Wahl dessen Parameter ist.

Im zweiten Teil dieser Dissertation betrachten wir das Problem, verrauschte Beobachtungen stationärer, ergodischer Zufallsprozesse endlicher Länge gemäss ihren generativen Modellen zu clustern, ohne Wissen über die Modellstatistiken und die Anzahl der generativen Modelle. Zwei Algorithmen, die beide den L^1-Abstand zwischen geschätzten Leistungsspektraldichten (englisch: power spectral densities (PSDs)) als Unähnlichkeitsmass verwenden, werden analysiert. Der erste Algorithmus, den wir nearest neighbor process clustering (NNPC) nennen, beruht auf der Partitionierung des Nächste-Nachbarn-Graphen der Beobachtungen mittels spektralem Clustern. Der zweite Algorithmus besteht aus einer einzigen k-Means-Iteration mit farthest point Initialisierung und wurde in der Literatur bereits behandelt, allerdings mit einem anderen Unähnlichkeitsmass und nur mit asymptotischer statistischer Analyse. Wir beweisen, dass beide Algorithmen in der Gegenwart von Rauschen und fehlenden Einträgen das richtige Ergebnis liefern, sogar wenn die generativen PSDs signifikant überlappen, sofern die Beobachtungslänge ausreichend gross ist. Unsere

Ergebnisse beschreiben die Wechselbeziehung zwischen dem Grad der Überlappung der generativen PSDs, der Beobachtungslänge, dem Anteil fehlender Einträge und dem Rauschpegel. Ausserdem präsentieren wir umfangreiche numerische Ergebnisse für synthetische und reale Daten. Diese Ergebnisse zeigen, dass NNPC die Cluster-Genauigkeit bekannter Algorithmen im Clustern von menschlicher Bewegungssequenzen übertrifft.

Im dritten Teil dieser Dissertation definieren wir das Problem der verteilungserhaltenden, verlustbehafteten Komprimierung (englisch: distribution-preserving lossy compression (DPLC)). Motiviert durch jüngste Fortschritte im Bereich der extremen Bildkomprimierung, die es erlauben, artefaktfreie Rekonstruktionen auch bei sehr niedrigen Bitraten aufrechtzuerhalten, schlagen wir vor, die Rate-Verzerrungs-Funktion unter der Nebenbedingung zu optimieren, dass die Rekonstruktionen der statistischen Verteilung von Trainingsdaten folgen. Ein solches Komprimierungssystem deckt ein breites Spektrum ab: Einerseits lernt es ein generatives Modell der Daten bei einer Bitrate von Null, und andererseits erreicht es perfekte Rekonstruktion bei genug hohen Bitraten. Darüber hinaus interpoliert es zwischen dem Lernen eines generativen Modells und der perfekten Rekonstruieren der Daten für mittelgrosse Bitraten. Wir untersuchen mehrere Methoden, um das vorgeschlagene Optimierungsproblem zu lösen, einschliesslich einer neuartigen Kombination von Wasserstein generative adversarial networks und Wasserstein autoencoders, und präsentieren eine umfangreiche theoretische und empirische Charakterisierung des vorgeschlagenen Komprimierungssystems.

Der vierte und letzte Teil dieser Dissertation befasst sich mit der hardwarefreundlichen Komprimierung tiefer neuronaler Netzwerke (englisch: deep neural networks (DNNs)) so, dass die komprimierten Modelle nur wenige Multiplikationen benötigen um Vorhersagen zu machen. Wir lernen Approximationen der (verallgemeinerten) Matrixmultiplikationen in DNN-Schichten, indem wir die Matrixmultiplikationen als 2-Schicht-Summen-Produkt-Netzwerke (SPN) (arithmetische Schaltungen) darstellen und deren (ternäre) Kantengewichte aus Daten lernen. Die SPN trennen (skalare) Multiplikations- und

Additionsoperationen und ermöglichen es uns, die Anzahl der verwendeten Multiplikationsoperationen festzulegen. Wir kombinieren unsere Methode mit knowledge distillation Techniken und wenden sie auf Bildklassifizierungs- und Sprachmodellierungs-DNNs an, und erhalten eine drastische Reduktion der Anzahl Multiplikationen (über 99,5%) unter Beibehaltung der Genauigkeit der ursprünglichen Modelle. Schliesslich demonstrieren wir, dass die vorgeschlagene Methode in der Lage ist, den Matrix-Multiplikationsalgorithmus von Strassen aus Beispielen zu lernen, d.h. das Multiplizieren von 2×2-Matrizen zu lernen unter der Verwendung von nur 7 Multiplikationen anstelle von 8.

Acknowledgments

I would like to express my deepest gratitude to my advisor, Professor Helmut Bölcskei, for his excellent guidance, advice, and support throughout my doctoral studies.

I would like to thank Professor Helmut Bölcskei and Professor Andreas Krause for acting as examiners for this thesis.

I would also like to thank Eiríkur Ágústsson, Professor Animashree Anandkumar, Aran Khanna, and Dr. Mario Lučić for discussions and contributions that were valuable for the outcome of my work.

Thanks go to the members of the Communication Theory Group for the enjoyable times we shared.

Finally, I would like to thank my family and friends for their encouragement and support over the years.

Contents

1 Introduction 1
1.1 Noisy subspace clustering via matching pursuits 1
1.2 Nearest neighbor random process clustering 6
1.3 Distribution-preserving lossy compression 8
1.4 Deep learning with a multiplication budget 11
1.5 Publications 13
1.6 Notation 14

2 Noisy subspace clustering via matching pursuits 17
2.1 Subspace clustering via matching pursuits 18
2.1.1 The algorithms 18
2.1.2 Parameter selection 22
2.2 Main results 27
2.3 Numerical results 36
2.3.1 Comparison of SSC-OMP and SSC-MP 37
2.3.2 Face clustering 40
2.3.3 True positives/false positives tradeoff in DD-stopping 42
2.3.4 Influence of $s_{\max}$ and $p_{\max}$ in DI-stopping . . . 45
2.3.5 Influence of $s_{\max}$ and $p_{\max}$ in DI-stopping for noiseless data 48
Appendices 50
2.A Proof of Theorem 2.1 50
2.B Proof of Theorem 2.2 73

2.C Proof of Theorem 2.3 81
2.D Supplementary notes 83

3 Nonparametric nearest neighbor random process clustering 85
3.1 Formal problem statement and algorithms 86
3.2 Analytical performance results 92
3.3 Comparison with thresholding-based subspace clustering 97
3.4 Numerical results 100
3.4.1 Synthetic data 100
3.4.2 Real data 107
Appendices 110
3.A Proofs of Theorems 3.1 and 3.2 110
3.B Proof of Theorem 3.3 112
3.C Proof of Proposition 3.1 123

4 Distribution-preserving lossy compression 127
4.1 Problem formulation 127
4.2 Distribution-preserving lossy compression 129
4.3 Unsupervised training via Wasserstein++ 133
4.4 Empirical evaluation 136
4.5 Related work 142
4.6 Conclusion and future work 144
Appendices 145
4.A Proof of Theorem 4.1 145
4.B Hyperparameters and architectures 146
4.C The Wasserstein++ algorithm 150
4.D Visual examples 151

5 Deep learning with a multiplication budget 165
5.1 Related work 165
5.2 Learning fast matrix multiplications via SPNs 167
5.2.1 Casting matrix multiplication as SPN 167
5.2.2 Learning fast approximate matrix multiplications for DNNs 169
5.2.3 Knowledge distillation (KD) 171

5.2.4 Application to 2D convolution 172
5.3 Experiments 174
5.3.1 Rediscovering Strassen's algorithm 174
5.3.2 Image classification 175
5.3.3 Language modeling 182
5.4 Conclusion and future work 184
Appendices 185
5.A Additional results on CIFAR-10 185
5.B Additional results for language modeling 186
5.C Application to 2D convolution: Pseudocode 186
5.D Additional tables 188

References 191

CHAPTER 1

Introduction

The first two parts of this thesis address clustering problems, namely subspace clustering and random process clustering. The third and fourth part deal with compression, specifically with deep generative models for lossy compression and neural network model compression for inference on resource-constrained hardware. Each part is introduced individually below.

1.1. NOISY SUBSPACE CLUSTERING VIA MATCHING PURSUITS (CHAPTER 2)

Extracting structural information from large high-dimensional data sets in a computationally efficient manner is a major challenge in many modern machine learning tasks. A structure widely encountered in practical applications is that of unions of (low-dimensional) subspaces. The problem of extracting the assignments of the data points in a given data set to the subspaces without prior knowledge of the number of subspaces, their orientations and dimensions is referred to as subspace clustering and has found applications in, e.g., image representation and segmentation (Hong et al., 2006), face clustering (Ho et al., 2003), motion segmentation (Costeira and Kanade, 1998), system identification (Vidal et al., 2003), and genomic inference (Jiang et al., 2004). More formally, given a set $\mathcal{Y} = \mathcal{Y}_1 \cup \ldots \cup \mathcal{Y}_L$ of N data points in $\mathbb{R}^m$, where the points in $\mathcal{Y}_\ell$ lie in or near the d_ℓ-dimensional linear subspace $\mathcal{S}_\ell \subset \mathbb{R}^m$, we want to find the association of the points in $\mathcal{Y}$ to the $\mathcal{Y}_\ell$, without prior knowledge on the $\mathcal{S}_\ell$.

The subspace clustering problem has been studied for more than two decades with a correspondingly sizeable body of literature. The algorithms available to date can roughly be categorized as algebraic, statistical, and spectral clustering-based; we refer to (Vidal, 2011) for a review of the most prominent representatives of each class. While many subspace clustering algorithms exhibit good performance in practice, corresponding analytical results under non-restrictive conditions on the relative orientations of the subspaces are available only for a small set of algorithms. Specifically, during the past few years a number of new algorithms, which rely on sparse representations (of each data point in terms of all the other data points) followed by spectral clustering (von Luxburg, 2007), were proposed and mathematically analyzed (Elhamifar and Vidal, 2013; Soltanolkotabi and Candès, 2012; Soltanolkotabi et al., 2014; Wang and Xu, 2016; Heckel and Bölcskei, 2015; Dyer et al., 2013; Park et al., 2014; You et al., 2016). These algorithms exhibit good empirical performance and succeed provably under quite generous conditions on the relative orientations of the subspaces. Almost all analytical performance results available to date apply, however, to the noiseless case, where the data points lie exactly in the union of the $\mathcal{S}_\ell$. A notable exception is the sparse subspace clustering (SSC) algorithm by Elhamifar and Vidal (2013), which was shown by Soltanolkotabi et al. (2014) and Wang and Xu (2016) to succeed for noisy data even when the subspaces intersect. SSC employs the Lasso[1] (or ℓ_1-minimization in the noiseless case) to find a sparse representation (or, more precisely, approximation) of each data point in terms of all the other data points, then constructs an affinity graph based on the so-obtained sparse representations, and

[1] We note that the SSC formulation in (Elhamifar and Vidal, 2013) adds a term to the Lasso objective function to account for sparse corruptions of the data points. The performance guarantees in (Soltanolkotabi et al., 2014; Wang and Xu, 2016) apply, however, to the "pure" Lasso version of SSC. Throughout this chapter, unless explicitly stated otherwise, SSC will refer to the "pure" Lasso version.

finally determines subspace assignments through spectral clustering of the affinity graph. To understand the intuition behind this approach, first note that in the noiseless case every data point $\mathbf{y}_j$ in $\mathcal{S}_\ell$ can be represented by (at most d_ℓ) other data points in $\mathcal{S}_\ell$ provided that the points in $\mathcal{Y}_\ell$ are non-degenerate. In the noisy case, the hope is now that the sparse representation of $\mathbf{y}_j \in \mathcal{Y}_\ell$ in terms of $\mathcal{Y}\backslash\{\mathbf{y}_j\}$ delivered by SSC involves mostly points belonging to $\mathcal{Y}_\ell$ thanks to the sparsity-promoting nature of the Lasso. Of course, this will happen only if the subspaces $\mathcal{S}_\ell$ underlying the $\mathcal{Y}_\ell$ are sufficiently far apart. The analytical performance results in (Soltanolkotabi and Candès, 2012; Soltanolkotabi et al., 2014; Wang and Xu, 2016) quantify the impact of subspace dimensions and relative orientations, noise variance, and the number of data points on the performance of SSC.

When the data is high-dimensional or the number of data points is large, solving the N Lasso problems (each in $N-1$ variables) in SSC can be computationally challenging. Greedy algorithms for computing sparse representations of the data points (in terms of all the other data points) are therefore an interesting alternative. Three such alternatives were proposed in the literature, namely the SSC-orthogonal matching pursuit (SSC-OMP) algorithm by Dyer et al. (2013), the thresholding-based subspace clustering (TSC) algorithm by Heckel and Bölcskei (2015), and the nearest subspace neighbor (NSN) algorithm by Park et al. (2014). SSC-OMP employs OMP instead of the Lasso to compute sparse representations of the data points. TSC relies on the nearest neighbors—in spherical distance—of each data point to construct the affinity graph, and NSN greedily assigns to each data point a subset of the other data points by iteratively selecting the data point closest (in Euclidean distance) to the subspace spanned by the previously selected data points.

To the best of our knowledge, besides SSC, TSC is the only subspace clustering algorithm that was proven to succeed under noise. The performance guarantees available for SSC-OMP (Dyer et al., 2013; You et al., 2016; Heckel et al., 2017) all apply to the noiseless case.

The main contributions of this chapter are an analytical performance characterization of SSC-OMP in the noisy case, and of a new algorithm, termed SSC-matching pursuit (SSC-MP), which is obtained by replacing OMP in SSC-OMP by the MP algorithm (Friedman and Stuetzle, 1981; Mallat and Zhang, 1993). Matching pursuit algorithms per se have been studied extensively in the sparse signal representation literature (Blumensath et al., 2012) and the approximation theory literature (Temlyakov, 2003). Replacing OMP by MP is attractive as the per-iteration complexity of MP is smaller than that of OMP thanks to the absence of the orthogonalization step. On the other hand, the representation error (in ℓ_2-norm) of MP may decay slower—as a function of the number of iterations—than that of OMP (Temlyakov, 2003). We shall see, however, that in the context of subspace clustering, in practice, the lower per-iteration cost of MP usually translates into lower overall running time, while delivering essentially the same clustering performance as OMP.

Our main results are sufficient conditions for SSC-OMP and SSC-MP to succeed in terms of the no false connections (NFC) property (see Definition 2.1), a widely used (Soltanolkotabi and Candès, 2012; Soltanolkotabi et al., 2014; Dyer et al., 2013; Heckel and Bölcskei, 2015; Wang and Xu, 2016; Heckel et al., 2017; Dyer et al., 2013; You et al., 2016; Park et al., 2014) subspace clustering performance measure. Specifically, we find that both algorithms succeed even when the subspaces intersect and when the signal to noise ratio is as low as 0dB. Furthermore, the sufficient conditions we obtain point at an intuitively appealing tradeoff between the affinity of the subspaces (a similarity measure for pairs of subspaces defined later), the noise variance, and the number of points in the data set corresponding to each subspace. This "clustering condition" is structurally similar to those for SSC in (Soltanolkotabi et al., 2014, Thm. 3.1), (Wang and Xu, 2016, Thm. 10) and for TSC in (Heckel and Bölcskei, 2015, Thm. 3). Moreover, numerical results indicate that our clustering condition is order-wise optimal. The main technical challenge in proving our results stems from the need to handle statistical dependencies between quantities computed in different iterations of the OMP and MP algorithms.

OMP and MP are commonly stopped either after a prescribed maximum number of iterations, which we henceforth call data-independent

(DI)-stopping, or when the representation error falls below a threshold value, referred to as data-dependent (DD)-stopping. For a given data point to be represented, OMP is guaranteed to select a new data point in every iteration and the sparsity level of the resulting representation therefore equals the number of OMP iterations performed. MP, on the other hand, may select individual data points to participate repeatedly in the sparse representation of a given data point. The sparsity level of the representation computed by MP may therefore be smaller than the number of iterations performed. As it is important for subspace clustering purposes to be able to control the sparsity level, we propose a new hybrid stopping criterion for MP terminating the algorithm either when a given maximum number of iterations was performed or when a given target sparsity level is attained. We consider SSC-OMP and SSC-MP both with DI- and DD-stopping. For DI-stopping, we present numerical results which indicate that performing (order-wise) more than d_ℓ OMP iterations can severely compromise the performance of SSC-OMP. SSC-OMP with DI-stopping therefore requires fairly accurate knowledge of the subspace dimensions. SSC-MP, on the other hand, exhibits a much more robust behavior in this regard. For DD-stopping, we prove that taking the threshold value on the representation error to be linear in the noise standard deviation ensures that both OMP and MP select order-wise at least d_ℓ points from $\mathcal{Y}_\ell \backslash \{\mathbf{y}_j\}$ to represent $\mathbf{y}_j \in \mathcal{Y}_\ell$, provided that the noise variance is sufficiently small. Numerical results further indicate that both algorithms, indeed, select order-wise no more than d_ℓ points from $\mathcal{Y}_\ell \backslash \{\mathbf{y}_j\}$ and essentially no points from $\mathcal{Y} \backslash \mathcal{Y}_\ell$. This means that SSC-OMP and SSC-MP with DD-stopping implicitly estimate (again order-wise) the subspace dimensions d_ℓ. This can—in principle—also be accomplished by SSC with a selection procedure for the Lasso parameter that is based on solving an auxiliary (constrained Lasso) optimization problem for each data point (Soltanolkotabi et al., 2014). This procedure imposes, however, significant computational burden; in contrast DD-stopping as performed here comes at essentially zero computational cost.

Finally, we present extensive numerical results comparing the performance of SSC-OMP, SSC-MP, SSC, TSC, and NSN for synthetic and real data. In particular, we find that SSC-MP outperforms SSC in the reference problem of face clustering on the Extended Yale B data set (Georghiades et al., 2001; Lee et al., 2005) and does so at drastically lower running time.

1.2. ROBUST NONPARAMETRIC NEAREST NEIGHBOR RANDOM PROCESS CLUSTERING (CHAPTER 3)

Consider a set of N noisy length-M observations of stationary ergodic discrete-time random processes stemming from $L < N$ (typically $L \ll N$) different generative processes, referred to as generative models henceforth. We want to cluster these observations according to their generative models without prior knowledge of the model statistics and the number of generative models, L. This problem arises in many domains of science and engineering where (large amounts of) data have to be divided into meaningful categories in an unsupervised fashion. Concrete examples include audio and video sequences (Wang et al., 2000), electrocardiography (ECG) recordings (Kalpakis et al., 2001), industrial production indices (Corduas and Piccolo, 2008), and financial time series (Marti et al., 2016a, 2017).

Common measures for quantifying the (dis)similarity of generative models typically rely on process statistics estimated from observations using either parametric or nonparametric methods. Parametric methods yield good performance when the (parametric) model the estimation is based on matches the true (unknown) model well. Nonparametric methods typically outperform parametric ones in case of model mismatch (Stoica and Moses, 2005), a likely scenario in many practical applications. Existing random process clustering methods quantify the dissimilarity of observations using the Euclidean distance between estimated process model parameters (Corduas and Piccolo, 2008; Marti et al., 2016a), cepstral coefficients (Kalpakis et al., 2001; Boets et al., 2005), or normalized periodograms (Caiado et al., 2006). Other methods rely on divergences (e.g., Kullback-Leibler divergence) between normalized periodograms (Kakizawa et al., 1998; Vilar and Pértega, 2004), use the distributional distance (Gray, 2009) between processes (Ryabko, 2010; Khaleghi et al., 2012, 2016), or the earth mover's distance between copulas of the processes (Marti et al., 2016a,b). In all cases the resulting distances are fed into a standard clustering algorithm such as k-means or hierarchical clustering. Another line of work employs a Bayesian framework to infer the cluster assignments, e.g., according to a maximum a posteriori criterion (Xiong and Yeung, 2004). While many of these approaches have proven effective in practice, corresponding analytical performance results are scarce. Moreover, existing analytical results are mostly concerned with the asymptotic regime where the observation length goes to infinity while the number of observations is fixed (see, e.g., (Kakizawa et al., 1998; Vilar and Pértega, 2004; Corduas and Piccolo, 2008; Ryabko, 2010; Khaleghi et al., 2012; Borysov et al., 2014)); the finite observation-length regime has attracted significantly less attention (Ryabko, 2010; Ryabko and Mary, 2013; Khaleghi et al., 2016; Marti et al., 2016a).

In this chapter, we consider two process clustering algorithms that apply to nonparametric generative models and employ the L^1-distance between estimated power spectral densities (PSDs) as dissimilarity measure. The first one, termed nearest neighbor process clustering (NNPC), relies on partitioning the q-nearest neighbor graph (q is a parameter of the algorithm) of the observations via normalized spectral clustering. NNPC is inspired by the TSC (subspace clustering) algorithm (Heckel and Bölcskei, 2015) and, to the best of our knowledge, has not been considered before in the context of process clustering. The second algorithm, which will be referred to as k-means (KM), consists of a single k-means iteration with farthest point initialization (Katsavounidis et al., 1994) and was first proposed in (Ryabko, 2010), albeit with a different dissimilarity measure.

Assuming real-valued stationary ergodic Gaussian processes with arbitrary (continuous) PSDs as generative models, we characterize the performance of NNPC and KM analytically for finite-length observations—potentially with missing entries—contaminated by independent additive real-valued white Gaussian noise. We find that both algorithms succeed with high probability even when the PSDs of the generative models exhibit significant overlap, all provided that the observation length is sufficiently large and the noise variance is sufficiently small. Our analytical results quantify the tradeoff between observation length, fraction of missing entries, noise variance, and distance between the (true) PSDs of the generative models.

Furthermore, we prove that treating the finite-length observations as vectors in Euclidean space and clustering them using the TSC algorithm (Heckel and Bölcskei, 2015) results in performance strictly inferior to that obtained for NNPC. We argue that the underlying cause is to be found in TSC employing spherical distance as dissimilarity measure, thereby ignoring the stationary process structure of the observations. In a broader context this suggests that clustering observations of random processes using dissimilarity measures conceived with Euclidean geometry in mind, a popular ad-hoc approach in practice (Esling and Agon, 2012), can lead to highly suboptimal performance.

We evaluate the performance of NNPC and KM on synthetic and on real data, and find that NNPC outperforms state-of-the-art algorithms in human motion sequence clustering. Furthermore, NNPC and KM are shown to yield better clustering performance than single linkage and average linkage hierarchical clustering based on the L^1-distance between estimated PSDs. We also compare (L^1-based) NNPC and KM to their respective L^2 and L^∞-cousins and find that the original variants consistently yield better or the same results.

1.3. DEEP GENERATIVE MODELS FOR DISTRIBUTION-PRESERVING LOSSY COMPRESSION (CHAPTER 4)

Data compression methods based on deep neural networks (DNNs) have recently received a great deal of attention. These methods were shown to outperform traditional codecs in image compression (Toderici et al., 2015, 2017; Theis et al., 2017; Rippel and Bourdev, 2017; Ballé et al., 2017; Agustsson et al., 2017; Johnston et al., 2017; Li et al., 2018; Mentzer et al., 2018; Ballé et al., 2018), speech compression (Kankanahalli, 2018), and video compression (Wu et al., 2018) under several distortion measures. In addition, DNN-based compression methods are flexible and can be adapted to specific domains leading to further reductions in bitrate, and promise fast processing thanks to their internal representations that are amenable to modern data processing pipelines (Torfason et al., 2018).

In the context of image compression, learning-based methods arguably excel at low bitrates by learning to realistically synthesize local image content, such as texture. While learning-based methods can lead to larger distortions w.r.t. measures optimized by traditional compression algorithms, such as peak signal-to-noise ratio (PSNR), they avoid artifacts such as blur and blocking, producing visually more pleasing results (Toderici et al., 2015, 2017; Theis et al., 2017; Rippel and Bourdev, 2017; Ballé et al., 2017; Agustsson et al., 2017; Johnston et al., 2017; Li et al., 2018; Mentzer et al., 2018; Ballé et al., 2018). In particular, visual quality can be improved by incorporating generative adversarial networks (GANs) (Goodfellow et al., 2014) into the learning process (Rippel and Bourdev, 2017; Agustsson et al., 2018). Rippel and Bourdev (2017) leveraged GANs for artifact suppression, whereas Agustsson et al. (2018) used them to learn synthesizing image content beyond local texture, such as facades of buildings, obtaining visually pleasing results at very low bitrates.

In the third chapter of this thesis, we propose a formalization of this line of work: *A compression system that respects the distribution of the original data at all rates*—a system whose decoder generates i.i.d. samples from the data distribution at zero bitrate, then gradually produces reconstructions containing more content of the original image as the bitrate increases, and eventually achieves perfect reconstruction at high enough bitrate (see Figure 4.1 for examples). Such a system can be learned from data in a fully unsupervised fashion by solving what we call the *distribution-preserving lossy compression (DPLC)* problem: Optimizing the rate-distortion tradeoff under the constraint that the reconstruction follows the distribution of the training data. Enforcing this constraint promotes artifact-free reconstructions, at all rates.

We then show that the algorithm proposed in (Agustsson et al., 2018) is solving a special case of the DPLC problem, and demonstrate that it fails to produce stochastic decoders as the rate tends to zero in practice, i.e., it is not effective in enforcing the distribution constraint at very low bitrates. This is not surprising as it was designed with a different goal in mind. We then propose and study different alternative approaches based on deep generative models that overcome the issues inherent with (Agustsson et al., 2018). In a nutshell, one first learns a generative model and then applies it to learn a stochastic decoder, obeying the distribution constraint on the reconstruction, along with a corresponding encoder. To quantify the distribution mismatch of the reconstructed samples and the training data in the learning process we rely on the Wasserstein distance. One distinct advantage of our approach is that we can theoretically characterize the distribution of the reconstruction and bound the distortion as a function of the bitrate.

On the practical side, to learn the generative model, we rely on Wasserstein generative adversarial network (WGAN) (Arjovsky et al., 2017) and Wasserstein autoencoder (WAE) (Tolstikhin et al., 2018), as well as a novel combination thereof termed Wasserstein++. The latter attains high sample quality comparable to WGAN (when measured in terms of the Fréchet inception distance (FID) (Heusel et al., 2017)) and yields a generator with good mode coverage as well as a structured latent space suited to be combined with an encoder, like WAE. We present an extensive empirical evaluation of the proposed approach on two standard GAN data sets, CelebA (Liu et al., 2015b) and LSUN bedrooms (Yu et al., 2015), realizing the first system that effectively solves the DPLC problem.

1.4. DEEP LEARNING WITH A MULTIPLICATION BUDGET (CHAPTER 5)

The outstanding predictive performance of DNNs often comes at the cost of large model size, and corresponding computational inefficiency. This can make the deployment of DNNs on mobile and embedded hardware challenging. For example, a full-precision ResNet-152 (He et al., 2016a) contains 60.2 million parameters and one forward pass requires 11.3 billion floating point operations. A variety of methods to address this issue were proposed recently, including optimizing the network architecture, factorizing the weight tensors, pruning the weights, and reducing the numerical precision of weights and activations (see Section 5.1 for a detailed overview).

These prior works mainly focused on decreasing the number of multiply-accumulate operations used by DNNs. In contrast, in this chapter, the objective that guides our algorithm design is a *reduction of the number of multiplications*. This algorithm design principle has led to many fast algorithms in linear algebra, most notably Strassen's matrix multiplication algorithm (Strassen, 1969). Strassen's algorithm uses 7 instead 8 multiplications to compute the product of two 2×2 matrices (and requires $O(n^{2.807})$ operations for multiplying $n \times n$ matrices). In the context of DNNs, the same design principle led to the Winograd filter-based convolution algorithm proposed by Lavin and Gray (2016). This algorithm only requires 16 instead of 36 multiplications to compute 2×2 outputs of 2D convolutions with 3×3 kernels and achieves a 2–3× speedup on graphics processing units (GPUs) in practice.

From a hardware perspective, multipliers occupy considerably more area on chip than adders (for fixed-point data types). Field-programmable gate arrays (FPGAs) and application-specific integrated circuits (ASICs) can therefore potentially accommodate considerably more adders than multipliers, and trading off multiplications against additions is desirable. In fact, it was demonstrated recently that DNN architectures which rely on a large number of additions and a small number of multiplications (such as (Li et al., 2016)) achieve a 60% higher throughput on FPGA than on GPU, while being 2.3× better in performance per watt (Nurvitadhi et al., 2017). In the context of ASICs, reducing the number of multiplications is beneficial as multiplication operations consume significantly more energy than addition operations (3–30× depending on the data type (Horowitz, 2014; Andri et al., 2018)). More generally, replacing multiplications in DNNs by additions leads to a reduction in models size as addition/subtraction can be encoded as a binary weight. This is beneficial in terms of throughput for most deep learning applications, which are typically memory-bound.

Motivated by these observations, we propose a novel framework to drastically reduce the number of multiplications used by DNNs for inference. Specifically, for every DNN layer, we cast the (matrix) multiplication of the weight matrix with the activations as a 2-layer sum-product network (SPN) (arithmetic circuit). The SPNs disentangle (scalar) multiplications and additions in a way similar to Strassen's algorithm. The number of hidden units in the SPNs therefore determines the multiplication budget of the corresponding DNN layers. We then *learn the addition and multiplication operations for all layers jointly from data* by learning the edges of the SPNs, encoded as ternary $\{-1, 0, 1\}$ matrices. As the transforms realized by the SPNs are approximate and adapted to the weight matrices and distribution of the activation tensors in the DNN, this allows us to reduce the number of multiplications much more drastically than hand-engineered transforms like Strassen's algorithm or the more specialized Winograd filter-based convolution. In summary, our main contributions are the following.

- We propose a SPN-based framework for stochastic gradient-based end-to-end learning of fast approximate transforms for the arithmetic operations in DNN layers.

- Our framework allows fine-grained control of the number of multiplications and additions used at inference time, enabling precise adjustment of the tradeoff between arithmetic complexity and accuracy of DNN models.

- Extensive evaluations on the CIFAR-10 and ImageNet data sets show that our method applied to ResNet (He et al., 2016a) yields the same or higher accuracy than existing complexity reduction methods while using considerably fewer multiplications. For example, for ResNet-18 our method reduces the number of multiplications by 99.63% while incurring a top-1 accuracy degradation of only 2.0% compared to the full-precision model on ImageNet.
- Our method applied to a language model with convolution and LSTM layers (Kim et al., 2016a) results in a 99.69% reduction in multiplications while inducing an increase of only 3.3% in perplexity.
- Combining our method with knowledge distillation (KD) techniques, we obtain for the first time massive reductions in number of multiplications (99.5% and more) while maintaining the predictive performance of the full-precision models, for both image classification and language modeling.
- We demonstrate that the proposed framework is able to rediscover Strassen's algorithm, i.e., it can learn to (exactly) multiply 2×2 matrices using only 7 multiplications instead of 8.

Two key aspects of our approach that lead to gains compared previous methods are (i) our method is specifically tailored to reduce the number of multiplications whereas some previous works put more emphasis on model size reduction, and (ii) we leverage knowledge distillation which improves our results further.

1.5. PUBLICATIONS

The majority of the results in this thesis have been published during the course of the PhD studies. Specifically, the results in Chapters 2 and 3 appear in (Tschannen and Bölcskei, 2018) and (Tschannen and

Bölcskei, 2015, 2017), respectively. Moreover, the results presented in Chapter 4 were accepted for publication (Tschannen et al., 2018a), and the results in Chapter 5 were published in (Tschannen et al., 2018b). Other publications relevant for this thesis are (Tschannen, 2014; Heckel et al., 2017; Locatello et al., 2017a,b; Agustsson et al., 2017; Torfason et al., 2018; Mentzer et al., 2018; Agustsson et al., 2018; Furlanello et al., 2018).

1.6. NOTATION

We use lowercase boldface letters to denote (column) vectors and uppercase boldface letters to designate matrices. The superscript $^\top$ stands for transposition. For the vector $\mathbf{v}$, $[\mathbf{v}]_i$ denotes its ith element, $\|\mathbf{v}\|_0$ is the number of non-zero entries, and $\|\mathbf{v}\|_\infty := \max_i |[\mathbf{v}]_i|$. For the matrix $\mathbf{A}$, $\mathbf{A}_{i,j}$ denotes the entry in the ith row and jth column, $\mathbf{A}_i$ designates its ith row, $\mathbf{A}_{-i}$ stands for the matrix obtained by removing the ith column from $\mathbf{A}$, $\mathbf{A}_{\mathcal{T}}$ is the submatrix of $\mathbf{A}$ consisting of the columns with index in the set $\mathcal{T}$, $\mathcal{R}(\mathbf{A})$ is its range space, $\|\mathbf{A}\|_{2\to 2} := \max_{\|\mathbf{v}\|_2=1} \|\mathbf{A}\mathbf{v}\|_2$ its spectral norm, $\|\mathbf{A}\|_F := (\sum_{i,j} |\mathbf{A}_{i,j}|^2)^{1/2}$ its Frobenius norm, $\operatorname{tr}(\mathbf{A}) = \sum_i \mathbf{A}_{i,i}$ its trace (for $\mathbf{A}$ square), and $\sigma_{\min}(\mathbf{A})$ and $\sigma_{\max}(\mathbf{A})$ refer to its minimum and maximum singular value, respectively. For a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, $m \geq n$, of full column rank, we denote its pseudoinverse by $\mathbf{A}^\dagger := (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top$. $\mathbf{I}$ and $\mathbf{1}$ stand for the identity matrix and the all ones matrix (the latter not necessarily square), respectively. $\operatorname{vec}(\mathbf{A})$ is the vectorization of the matrix $\mathbf{A} = [\mathbf{a}_1 \ldots \mathbf{a}_n]$, i.e., $\operatorname{vec}(\mathbf{A}) = [\mathbf{a}_1{}^\top \ldots \mathbf{a}_n{}^\top]^\top$. For matrices $\mathbf{A}$ and $\mathbf{B}$ of identical dimensions, $\mathbf{A}\odot\mathbf{B}$ is the Hadamard product, i.e., $(\mathbf{A}\odot\mathbf{B})_{i,j} = \mathbf{A}_{i,j}\mathbf{B}_{i,j}$. For a vector $\mathbf{b} \in \{0,1\}^n$, we let $\mathbf{P}_\mathbf{b} := \operatorname{diag}([\mathbf{b}]_1, \ldots, [\mathbf{b}]_n)$.

The set $\{1,\ldots,N\}$ is denoted by $[N]$. The cardinality of the set $\mathcal{T}$ is $|\mathcal{T}|$ and its complement is $\overline{\mathcal{T}}$. The unit sphere in $\mathbb{R}^m$ is $\mathbb{S}^{m-1} := \{\mathbf{x} \in \mathbb{R}^m \colon \|\mathbf{x}\|_2 = 1\}$.

The ith element of a sequence x is denoted by $x[i]$. The (circular) convolution of $f, g \in L^2([0,1))$ is defined as $(f * g)(y) := \int_0^1 f(x)\tilde{g}(y-x)\mathrm{d}x$, $y \in [0,1)$, where $\tilde{g}$ is the 1-periodic extension of

g. The composition of f and g is written as $f \circ g$. $\log(\cdot)$ refers to the natural logarithm.

$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ stands for the distribution of a Gaussian random vector with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The expectation of the random variable X is written as $\mathbb{E}[X]$ and its distribution is denoted by P_X. The relation $W \sim P_X$ designates that W follows the distribution P_X, and $X \sim Y$ indicates that X and Y are identically distributed.

We say that a subgraph H of a graph G is connected if every pair of nodes in H can be joined by a path with nodes exclusively in H. A connected subgraph H of G is called a connected component of G if there are no edges between H and the remaining nodes in G.

CHAPTER 2

Noisy Subspace Clustering via Matching Pursuits

Sparsity-based subspace clustering algorithms have attracted significant attention thanks to their excellent performance in practical applications. A prominent example is the SSC algorithm by Elhamifar and Vidal (2013), which performs spectral clustering based on an adjacency matrix obtained by sparsely representing each data point in terms of all the other data points via the Lasso. When the number of data points is large or the dimension of the ambient space is high, the computational complexity of SSC quickly becomes prohibitive. Dyer et al. (2013) observed that SSC-OMP obtained by replacing the Lasso by the greedy OMP algorithm results in significantly lower computational complexity, while often yielding comparable performance. The central goal of this chapter is an analytical performance characterization of SSC-OMP for noisy data. Moreover, we introduce and analyze the SSC-MP algorithm, which employs MP in lieu of OMP. Both SSC-OMP and SSC-MP are proven to succeed even when the subspaces intersect and when the data points are contaminated by severe noise. The clustering conditions we obtain for SSC-OMP and SSC-MP are similar to those for SSC and for the TSC algorithm due to Heckel and Bölcskei (2015). Analytical results in combination with numerical results indicate that both SSC-OMP and SSC-MP with a data-dependent stopping criterion automatically detect the dimensions of the subspaces underlying the data. Experiments on synthetic and on real data show that SSC-MP often matches or exceeds the performance of the computationally more expensive SSC-OMP algorithm. Moreover, SSC-MP compares very favorably to SSC, TSC, and the NSN algorithm, both in terms of clustering performance and running time. In addition, we find that, in contrast to SSC-OMP, the performance of SSC-MP is very robust with respect to the choice of parameters in the stopping criteria.

2.1. SUBSPACE CLUSTERING VIA MATCHING PURSUITS

OMP and MP per se were introduced in (Chen et al., 1989) and (Friedman and Stuetzle, 1981), respectively, and have been studied extensively in the sparse signal representation literature, see, e.g., (Blumensath et al., 2012), (Foucart and Rauhut, 2013, Chap. 3). In the context of subspace clustering the premise is that the (sparse) representations of each data point in terms of all the other data points delivered by OMP and MP contain predominantly data points that lie in the same subspace as the data point under consideration. We refer to (Elhamifar and Vidal, 2013), (Heckel and Bölcskei, 2015, Sec. 2.B) for a detailed discussion on the relation between sparse signal representation theory and subspace clustering.

2.1.1. The algorithms

We first briefly review the SSC-OMP algorithm, introduced in (Dyer et al., 2013), and then present the novel SSC-MP algorithm. The ensuing formulations of SSC-OMP and SSC-MP assume that the data points are of comparable ℓ_2-norm. This assumption is relevant in Step 1 in both algorithms, but is not restrictive as the data points can always be normalized prior to processing. Further, an estimate $\hat{L}$ of the number of subspaces L is assumed to be available. The estimation of L from the data set under consideration is discussed below.

The SSC-OMP algorithm (Dyer et al., 2013)**:** *Given a set of N data points $\mathcal{Y}$ in $\mathbb{R}^m$, an estimate of the number of subspaces $\hat{L}$, and*

- *a maximum number of iterations $s_{\max} \leq \min\{m, N-1\}$ for DI-stopping,*
- *a threshold τ on the representation error for DD-stopping,*

perform the following steps:

Step 1: *For every $\mathbf{y}_j \in \mathcal{Y}$, find a representation of $\mathbf{y}_j$ in terms of $\mathcal{Y} \backslash \{\mathbf{y}_j\}$ using OMP as follows: Initialize the iteration counter $s = 0$, the residual $\mathbf{r}_0 = \mathbf{y}_j$, and the set of selected indices $\Lambda_0 = \emptyset$. Denote the data matrix containing the points in $\mathcal{Y}$ by $\mathbf{Y} \in \mathbb{R}^{m \times N}$. For $s = 1, 2, \ldots$, perform the updates*

$$\begin{aligned} \lambda_s &= \operatorname*{arg\,max}_{i \in [N] \backslash (\Lambda_{s-1} \cup \{j\})} |\langle \mathbf{y}_i, \mathbf{r}_{s-1} \rangle| && \textit{(2.1)} \\ \Lambda_s &= \Lambda_{s-1} \cup \lambda_s \\ \mathbf{r}_s &= \left(\mathbf{I} - \mathbf{Y}_{\Lambda_s} (\mathbf{Y}_{\Lambda_s})^{\dagger} \right) \mathbf{y}_j && \textit{(2.2)} \\ &= \left(\mathbf{I} - \frac{\tilde{\mathbf{y}}_{\lambda_s} (\tilde{\mathbf{y}}_{\lambda_s})^{\top}}{\|\tilde{\mathbf{y}}_{\lambda_s}\|_2^2} \right) \mathbf{r}_{s-1}, \end{aligned}$$

where $\tilde{\mathbf{y}}_{\lambda_s} = (\mathbf{I} - \mathbf{Y}_{\Lambda_{s-1}} (\mathbf{Y}_{\Lambda_{s-1}})^{\dagger}) \mathbf{y}_{\lambda_s}$, until at least one of the following criteria is met

- *for DI-stopping:*
  $s = s_{\max}$, $\max_{i \in [N] \backslash (\Lambda_s \cup \{j\})} |\langle \mathbf{y}_i, \mathbf{r}_s \rangle| = 0.$[1]
- *for DD-stopping:*
  $\|\mathbf{r}_s\|_2 \leq \tau$, $\max_{i \in [N] \backslash (\Lambda_s \cup \{j\})} |\langle \mathbf{y}_i, \mathbf{r}_s \rangle| = 0.$

Ties in the maximization (2.1) *are broken arbitrarily.*

Step 2: *With the number of OMP iterations actually performed denoted by s_{a}, compute the representation coefficient vectors $\mathbf{b}_j \in \mathbb{R}^N$, $j \in [N]$, according to $(\mathbf{b}_j)_{\Lambda_{s_{\mathrm{a}}}} = (\mathbf{Y}_{\Lambda_{s_{\mathrm{a}}}})^{\dagger} \mathbf{y}_j$, $(\mathbf{b}_j)_{\overline{\Lambda_{s_{\mathrm{a}}}}} = \mathbf{0}$, and construct the adjacency matrix $\mathbf{A} = \mathbf{B} + \mathbf{B}^{\top}$, where $\mathbf{B} = \operatorname{abs}([\mathbf{b}_1 \ \ldots \ \mathbf{b}_N])$*

[1] Throughout the chapter, we use the convention of maximization over the empty set evaluating to 0.

with $\mathrm{abs}(\cdot)$ *denoting absolute values taken element-wise.*

Step 3: *Apply normalized spectral clustering (Ng et al., 2001; von Luxburg, 2007) to* $(\mathbf{A}, \hat{L})$*.*

The SSC-MP algorithm: *Given a set of* N *data points* $\mathcal{Y}$ *in* $\mathbb{R}^m$*, an estimate of the number of subspaces* $\hat{L}$*, and*

- *a maximum number of iterations* $s_{\max}$ *and a target sparsity level* $p_{\max}$ *for DI-stopping,*
- *a threshold* τ *on the representation error for DD-stopping,*

perform the following steps:

Step 1: *For every* $\mathbf{y}_j \in \mathcal{Y}$*, find a representation of* $\mathbf{y}_j$ *in terms of* $\mathcal{Y} \backslash \{\mathbf{y}_j\}$ *using MP as follows: Initialize the iteration counter* $s = 0$*, the residual* $\mathbf{q}_0 = \mathbf{y}_j$*, and the coefficient vector* $\mathbf{b}_j \in \mathbb{R}^N$ *as* $\mathbf{b}_j = \mathbf{0}$*. For* $s = 1, 2, \ldots,$ *perform the updates*

$$\omega_s = \underset{i \in [N] \backslash \{j\}}{\arg\max} \, |\langle \mathbf{y}_i, \mathbf{q}_{s-1} \rangle| \tag{2.3}$$

$$[\mathbf{b}_j]_{\omega_s} \leftarrow [\mathbf{b}_j]_{\omega_s} + \frac{\langle \mathbf{y}_{\omega_s}, \mathbf{q}_{s-1} \rangle}{\|\mathbf{y}_{\omega_s}\|_2^2} \tag{2.4}$$

$$\mathbf{q}_s = \left(\mathbf{I} - \frac{\mathbf{y}_{\omega_s} (\mathbf{y}_{\omega_s})^\top}{\|\mathbf{y}_{\omega_s}\|_2^2} \right) \mathbf{q}_{s-1} \tag{2.5}$$

until at least one of the following criteria is met

- *for DI-stopping:*
  $s = s_{\max}$*,* $\|\mathbf{b}_j\|_0 = p_{\max}$*,* $\max_{i \in [N] \backslash \{j\}} |\langle \mathbf{y}_i, \mathbf{q}_s \rangle| = 0$.
- *for DD-stopping:*
  $\|\mathbf{q}_s\|_2 \leq \tau$*,* $\max_{i \in [N] \backslash \{j\}} |\langle \mathbf{y}_i, \mathbf{q}_s \rangle| = 0$.

Ties in the maximization (2.3) *are broken arbitrarily.*

Step 2: *Construct the adjacency matrix* $\mathbf{A} = \mathbf{B} + \mathbf{B}^\top$*, where* $\mathbf{B} = \mathrm{abs}([\mathbf{b}_1 \, \ldots \, \mathbf{b}_N])$*.*

Step 3: *Apply normalized spectral clustering (Ng et al., 2001; von Luxburg, 2007) to* $(\mathbf{A}, \hat{L})$*.*

Stopping criteria: We emphasize that OMP, thanks to the orthogonalization (2.2) of the residual $\mathbf{r}_{s-1}$ w.r.t. all data points selected previously, is guaranteed to select a new data point in every iteration and hence the sparsity level of $\mathbf{b}_j$ equals the number of OMP iterations performed. In contrast, MP orthogonalizes (see (2.5)) the residual $\mathbf{q}_{s-1}$ w.r.t. the data point $\mathbf{y}_{\omega_s}$ selected in the current iteration s only and may therefore select the same data point to participate repeatedly in the representation of $\mathbf{y}_j$. The sparsity level of $\mathbf{b}_j$ may hence be smaller than the number of MP iterations performed, which is why the DI-stopping criterion for MP incorporates termination when a given target sparsity level, namely $p_{\max}$, is attained. Choosing $s_{\max}$ large enough, stopping will, indeed, be activated by $\|\mathbf{b}_j\|_0 = p_{\max}$. Having control over the sparsity level of the coefficient vectors $\mathbf{b}_j$ can be important to achieve good clustering performance as discussed below. Setting $p_{\max} = N$, on the other hand, guarantees that stopping is activated through $s = s_{\max}$ and thereby allows to control the maximum number of MP iterations through choice of $s_{\max}$. This hybrid stopping criterion does not seem to have been considered before in the literature.

For DD-stopping, OMP is guaranteed to stop as soon as a basis for the subspace $\mathbf{y}_j$ lies in has been found or, in case $\mathbf{y}_j$ does not lie in the span of $\mathcal{Y}\backslash\{\mathbf{y}_j\}$, the best representation—in the least-squares sense—of $\mathbf{y}_j$ in terms of the points in $\mathcal{Y}\backslash\{\mathbf{y}_j\}$. On the other hand, MP is, in general, not guaranteed to terminate after a finite number of iterations as it may fail to activate either of the conditions $\|\mathbf{q}_s\|_2 \leq \tau$ and $\max_{i\in[N]\backslash\{j\}} |\langle \mathbf{y}_i, \mathbf{q}_s\rangle| = 0$ if τ is chosen too small (Mallat and Zhang, 1993). For most data sets encountered in practice this is not an issue. It can, however, become a problem when the data set contains outliers that cannot be represented sparsely by the other data points. In such cases it is advisable to employ the DI-stopping criterion which guarantees that at most $s_{\max}$ iterations are performed.

Implementation aspects: As MP requires the computation of inner products only, whereas OMP contains a (least-squares) orthogonalization step (typically carried out by QR decomposition or Cholesky factorization (Blumensath et al., 2012)), the per-iteration computational cost of SSC-MP is lower than that of SSC-OMP. The numerical results in Section 2.3.2 indicate that this typically also translates into a lower overall running time for SSC-MP at fixed performance.

Weak selection rules: For very large data sets one can often speed up OMP and MP by relaxing the selection rules (2.1) and (2.3) to so-called weak selection rules (Blumensath et al., 2012) as follows. Instead of (2.1), one determines λ_s such that $|\langle \mathbf{y}_{\lambda_s}, \mathbf{r}_{s-1}\rangle| \geq \alpha \max_{i \in [N]\setminus(\Lambda_{s-1}\cup\{j\})} |\langle \mathbf{y}_i, \mathbf{r}_{s-1}\rangle|$ and instead of (2.3), one finds ω_s according to $|\langle \mathbf{y}_{\omega_s}, \mathbf{q}_{s-1}\rangle| \geq \alpha \max_{i\in[N]\setminus\{j\}} |\langle \mathbf{y}_i, \mathbf{q}_{s-1}\rangle|$, in both cases for a fixed relaxation parameter $\alpha \in (0, 1]$. These weakened selection rules can be implemented efficiently using, e.g., locality-sensitive hashing (Jain et al., 2011; Vitaladevuni et al., 2011). For conciseness, we shall not analyze weak selection rules here, but only note that our main results presented in Section 2.2 extend to weak selection rules with minor modifications.

2.1.2. Parameter selection

In both algorithms, $\hat{L}$ may be estimated in Step 2 based on the adjacency matrix $\mathbf{A}$ using the *eigengap heuristic* (von Luxburg, 2007) (note that L is needed only in Step 3), which relies on the fact that the number of zero eigenvalues of the normalized Laplacian of the graph G with adjacency matrix $\mathbf{A}$ corresponds to the number of connected components of G.

The spectral clustering step (Step 3 in both algorithms) recovers the oracle segmentation $\{\mathcal{Y}_1, \dots, \mathcal{Y}_L\}$ of $\mathcal{Y}$ if $\hat{L} = L$ and if each connected component in G corresponds to exactly one of the $\mathcal{Y}_\ell$ (von Luxburg, 2007, Prop. 4; Sec. 7). Choosing the parameters $s_{\max}$ and $p_{\max}$ in the case of DI-stopping, and τ for DD-stopping, appropriately is therefore crucial. Indeed, as $s_{\max}$, $p_{\max}$, and τ determine the sparsity level of the representation coefficient vectors $\mathbf{b}_j$, they control the number of edges in G and hence the connectivity properties of G. Specifically, taking $s_{\max}$, $p_{\max}$ too small or τ too large results in weak connectivity and may hence cause the subgraphs of G corresponding to individual $\mathcal{Y}_\ell$ to split up into multiple connected components. Spectral clustering would then assign these components to different clusters, which means that a given set $\mathcal{Y}_\ell$ is divided up into multiple (disjoint) sets. On the other hand, choosing $s_{\max}$, $p_{\max}$ too big or τ too small will result in strong connectivity and hence potentially in "false connections", i.e., in edges in G between points that correspond to different $\mathcal{Y}_\ell$. Spectral clustering exhibits, however, a certain amount of robustness vis-à-vis false connections with small corresponding weights in G. From what was just said it follows that ideally $s_{\max}$, $p_{\max}$, and τ should be chosen such that the sparsity levels of the $\mathbf{b}_j$ are on the order of the subspace dimensions. This can be seen as follows. Assume that the points in $\mathcal{Y}_\ell$ are well spread out on the subspace $\mathcal{S}_\ell$, $\ell \in [L]$, and perturbed by additive isotropic Gaussian noise. If the noise variance is not too large the noisy data points $\mathbf{y}_j \in \mathcal{Y}_\ell$ will remain close to $\mathcal{S}_\ell$. In this case, roughly d_ℓ points from $\mathcal{Y}_\ell \backslash \{\mathbf{y}_j\}$ will suffice to represent $\mathbf{y}_j \in \mathcal{Y}_\ell$ with small representation error. Hence, if the subspaces $\mathcal{S}_\ell$ are sufficiently far apart, imposing a sparsity level of $\approx d_\ell$ for $\mathbf{y}_j \in \mathcal{Y}_\ell$ through suitable choice of $s_{\max}$, $p_{\max}$, τ will force OMP and MP to select points predominantly from $\mathcal{Y}_\ell \backslash \{\mathbf{y}_j\}$. If, however, the subspaces are too close to each other, OMP and MP are likely to select points from $\mathcal{Y} \backslash \mathcal{Y}_\ell$ (i.e., false connections) as well.

We next discuss the selection of the parameters $s_{\max}$, $p_{\max}$ for DI- and τ for DD-stopping in the light of what was just said. For simplicity of exposition, in the noisy case we assume that the points in $\mathcal{Y}$ are in general position, i.e., every subset of m or fewer points in $\mathcal{Y}$ is linearly independent. Real-world data sets usually have this property and the statistical data model our analysis is based on (see Section 2.2) conforms with this assumption as well. Note that if the points in $\mathcal{Y}$ are in general position, OMP under DI-stopping is guaranteed to perform exactly $s_{\max}$ iterations, whereas MP will perform at least $\min\{s_{\max}, p_{\max}, m, N-1\}$ iterations under DI-stopping.

DI-stopping: We first consider SSC-OMP. In the noiseless case, if the subspaces are sufficiently far apart, OMP under DI-stopping automatically stops after at most d_ℓ iterations for all $\mathbf{y}_j \in \mathcal{Y}_\ell$, $\ell \in [L]$ (Dyer et al., 2013; You et al., 2016). Therefore, choosing $s_{\max} \geq \max_{\ell \in [L]} d_\ell$ guarantees that OMP automatically detects the dimensions of the subspaces the points $\mathbf{y}_j$ reside in. In contrast, in the noisy case, performing more than $\approx d_\ell$ iterations "forces" OMP to select points from $\mathcal{Y} \backslash \mathcal{Y}_\ell$ (corresponding to false connections) for the representation of $\mathbf{y}_j \in \mathcal{Y}_\ell$ as after $\approx d_\ell$ iterations $\mathbf{r}_s$ will roughly be orthogonal to $\mathcal{S}_\ell$ (see Figure 2.1). The choice of $s_{\max}$, in practice, therefore requires knowledge of the subspace dimensions. In certain applications such as, e.g., face clustering (Basri and Jacobs, 2003), information on the subspace dimensions may, indeed, be available a priori. When the subspace dimensions d_ℓ vary widely (across ℓ), there may be no $s_{\max}$ that ensures both connectivity of all the subgraphs of G corresponding to the $\mathcal{Y}_\ell$, and at the same time guarantees a small number of false connections. In principle this problem could be mitigated by setting $s_{\max}$ for each $\mathbf{y}_j$ individually according to the dimension of the subspace it belongs to. This would, however, require knowledge of the assignments of the data points $\mathbf{y}_j$ to the subspaces $\mathcal{S}_\ell$, thereby performing the actual subspace clustering task.

We now turn to SSC-MP. MP with $p_{\max} = N$, i.e., the hybrid stopping criterion is activated once $s_{\max}$ iterations have been performed, exhibits a stopping behavior that is fundamentally different from that of OMP. As already mentioned this is a consequence of MP orthogonalizing the residual only w.r.t. the data point selected in the current iteration, thereby allowing for repeated selection of individual data points. In the noiseless case, unlike SSC-OMP, SSC-MP can select more than d_ℓ data points from $\mathcal{Y}_\ell \backslash \{\mathbf{y}_j\}$ to represent $\mathbf{y}_j \in \mathcal{Y}_\ell$, thereby potentially producing a graph G with better connectivity than SSC-OMP. On the other hand, SSC-MP tends to assign smaller weights to the points in $\mathcal{Y}_\ell \backslash \{\mathbf{y}_j\}$ than SSC-OMP, which can lead to slightly inferior performance (see the experiments reported in Appendix 2.3.5). For the noisy case, the experiments on synthetic and real-world data, reported in Section 2.3.4, show that in the first d_ℓ or so iterations MP adds predominantly new points from the correct cluster $\mathcal{Y}_\ell \backslash \{\mathbf{y}_j\}$ and thereafter tends to revisit previously selected data points or add new

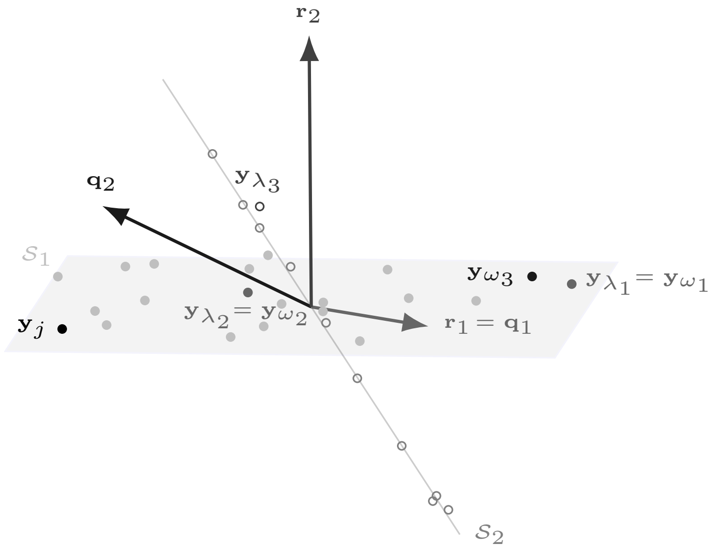

$\mathbf{r}_2$
$\mathbf{q}_2$
$\mathbf{y}_{\lambda_3}$
$\mathcal{S}_1$
$\mathbf{y}_{\omega_3}$
$\mathbf{y}_{\lambda_1} = \mathbf{y}_{\omega_1}$
$\mathbf{y}_j$
$\mathbf{y}_{\lambda_2} = \mathbf{y}_{\omega_2}$
$\mathbf{r}_1 = \mathbf{q}_1$
$\mathcal{S}_2$

Fig. 2.1: Evolution of the (normalized) residuals $\mathbf{r}_s$ and $\mathbf{q}_s$ corresponding to OMP and MP, respectively, for the data point $\mathbf{y}_j \in \mathcal{Y}_1$. The data points belonging to $\mathcal{S}_2$ are marked by circles. Note that all data points were normalized to unit ℓ_2-norm prior to clustering.

points stemming mostly from $\mathcal{Y}_\ell \backslash \{\mathbf{y}_j\}$. For fixed $s_{\max}$, this leads to a smaller number of false connections than what would be obtained by OMP. The example in Figure 2.1 illustrates this behavior. The data set $\mathcal{Y}$ is drawn from the union of a two-dimensional subspace $\mathcal{S}_1$ and a one-dimensional subspace $\mathcal{S}_2$, where $\mathcal{S}_1$ and $\mathcal{S}_2$ span a (principal) angle of approximately 50 degrees. We consider the data point $\mathbf{y}_j \in \mathcal{Y}_1$. In the first iteration, both OMP and MP select the same data point $\mathbf{y}_{\lambda_1} = \mathbf{y}_{\omega_1}$ and we have $\mathbf{r}_1 = \mathbf{q}_1$. In the second iteration, the two algorithms again select the same data point, namely $\mathbf{y}_{\lambda_2} = \mathbf{y}_{\omega_2}$, but $\mathbf{r}_2$ is now almost orthogonal to $\mathcal{S}_1$ whereas $\mathbf{q}_2$ remains close to $\mathcal{S}_1$ as MP orthogonalizes only w.r.t. $\mathbf{y}_{\omega_2}$. Specifically, the (principal) angle between $\mathbf{q}_2$ and $\mathcal{S}_1$ is approximately 27 degrees, while $\mathbf{q}_2$ and $\mathcal{S}_2$

span an angle of approximately 50 degrees. In the third iteration MP therefore selects a point from the set $\mathcal{Y}_1 \backslash \{\mathbf{y}_j\}$, whereas OMP chooses a point from the wrong subspace $\mathcal{Y}_2$.

The experiments reported in Section 2.3.4 reveal the following remarkable property. Even when we discount the advantage of MP—owing to its ability to reselect data points—by forcing the sparsity levels of MP and OMP to be equal to, say s_{high}, through appropriate choice of $s_{\max}$ and $p_{\max}$, with $s_{\text{high}} \gg d_\ell$ for at least one $\ell \in [L]$, MP still tends to select more points from $\mathcal{Y}_\ell$ to represent $\mathbf{y}_j \in \mathcal{Y}_\ell$ than OMP does. Moreover, these numerical results indicate that MP also tends to assign smaller (in absolute value) coefficients to false connections, i.e., to points in $\mathcal{Y} \backslash \mathcal{Y}_\ell$; this can have a favorable effect on performance thanks to the robustness of spectral clustering to false connections with small associated weights.

Our analytical results in Section 2.2 guarantee that SSC-OMP and SSC-MP succeed for $s_{\max}$ and $p_{\max}$ linear—up to log-terms—in the smallest subspace dimension, provided that the subspaces are sufficiently far apart, the noise variance is sufficiently small, and the data set contains sufficiently many points from each subspace.

DD-stopping: We assume throughout that τ is sufficiently large for SSC-MP to terminate. For OMP in the context of sparse noisy signal recovery, taking τ linear in the noise standard deviation σ is known to lead to correct recovery of the sparse signal under certain technical conditions (Cai and Wang, 2011). In the context of subspace clustering, where the problem is actually of a different nature, the analytical results in Section 2.2 indicate that such a choice for τ guarantees that both SSC-OMP and SSC-MP select order-wise at least d_ℓ points from $\mathcal{Y}_\ell$ to represent $\mathbf{y}_j \in \mathcal{Y}_\ell$, provided that the subspaces $\mathcal{S}_\ell$ are sufficiently far apart and the points in $\mathcal{Y}_\ell$ are well spread out on $\mathcal{S}_\ell$, $\ell \in [L]$, and perturbed by additive isotropic Gaussian noise of sufficiently small variance. The numerical results in Section 2.3.3 show that the graphs G generated by SSC-OMP and SSC-MP will also have a small number of false connections if τ, in addition, is not too small. Appropriate choice of τ therefore makes both OMP and MP automatically adjust the sparsity level for each data point according to the dimension of the subspace the data point lies in. We say that the algorithms detect the dimensions of the subspaces the data points reside in. Recall that under DI-stopping this has to be accomplished through suitable choice of $s_{\max}$, $p_{\max}$.

When the data set $\mathcal{Y}$ contains outliers that cannot be represented sparsely in terms of the other points in $\mathcal{Y}$, DD-stopping usually leads to a high number of false connections as the representation error for outliers will decay slowly resulting in late activation of the DD-stopping criterion. In this case, we need to rely on DI-stopping.

Summary: We recommend DI-stopping with $s_{\max}$ on the order of the subspace dimensions if the subspace dimensions are (approximately) known, and DD-stopping with τ^2 on the order of the noise variance if the noise variance is known and not too large. If no prior knowledge on the subspace dimensions or noise variance is available, or if the noise variance is large, we recommend relying on DI-stopping with $s_{\max}$ in the range $\{5, \ldots, 10\}$. These values for $s_{\max}$ turned out to work well in the relevant experiments conducted in this chapter, and were also used in related works (Dyer et al., 2013; You et al., 2016).

2.2. MAIN RESULTS

Our analytical performance results are for a statistical data model, also employed in (Soltanolkotabi et al., 2014; Heckel and Bölcskei, 2015). Specifically, we take the subspaces $\mathcal{S}_\ell$ to be fixed and the points in the corresponding subsets $\mathcal{Y}_\ell$ of the data set $\mathcal{Y} = \mathcal{Y}_1 \cup \ldots \cup \mathcal{Y}_L$ to be randomly distributed on $\mathcal{S}_\ell \cap \mathbb{S}^{m-1}$ and perturbed by additive random noise. Concretely, the points in $\mathcal{Y}_\ell$, $\ell \in [L]$, are given by $\mathbf{y}_i^{(\ell)} = \mathbf{x}_i^{(\ell)} + \mathbf{z}_i^{(\ell)} = \mathbf{U}^{(\ell)}\mathbf{a}_i^{(\ell)} + \mathbf{z}_i^{(\ell)}$, $i \in [n_\ell]$, where the columns of $\mathbf{U}^{(\ell)} \in \mathbb{R}^{m \times d_\ell}$ constitute an orthonormal basis for $\mathcal{S}_\ell$, the $\mathbf{a}_i^{(\ell)}$ are independently (across $i \in [n_\ell]$, $\ell \in [L]$) and uniformly distributed on $\mathbb{S}^{d_\ell - 1}$, and the $\mathbf{z}_i^{(\ell)}$ are i.i.d. $\mathcal{N}(\mathbf{0}, (\sigma^2/m)\mathbf{I}_m)$. The factor $1/m$ in the noise covariance matrix ensures that $\|\mathbf{z}_i^{(\ell)}\|_2^2$ concentrates around

$\mathbb{E}[\|\mathbf{z}_i^{(\ell)}\|_2^2] = \sigma^2$ for large m. Note further that the ℓ_2-norm of the data points concentrates around $\sqrt{1+\sigma^2}$ for large m and that they are hence of comparable ℓ_2-norm, as required by the formulations of SSC-OMP and SSC-MP in Section 2.1. Moreover, the data points are in general position w.p. 1 for $\sigma > 0$.

Prima facie assuming the noiseless data points $\mathbf{x}_i^{(\ell)}$ to be uniformly distributed on the subspaces $\mathcal{S}_\ell$ may appear overly stylized. However, for any algorithm to have a chance of producing correct assignments, we need the noiseless data points to be well spread out to a certain extent (albeit not necessarily around the origin as in our data model) on the subspaces. To see this, suppose for example, that the points in $\mathcal{Y}_\ell$ are concentrated on two distinct subspaces of $\mathcal{S}_\ell$, say $\mathcal{S}_\ell'$ and $\mathcal{S}_\ell''$. Then, one can assign the points in $\mathcal{Y}_\ell$ either to two clusters, one containing the points concentrated on $\mathcal{S}_\ell'$ and the other one those concentrated on $\mathcal{S}_\ell''$, or one can assign all the points in $\mathcal{Y}_\ell$ to a single cluster.

Our results will depend on the affinity between pairs of subspaces which measures how far apart two subspaces are. The affinity between the subspaces $\mathcal{S}_k$ and $\mathcal{S}_\ell$ is defined as (Soltanolkotabi and Candès, 2012, Def. 2.6), (Soltanolkotabi et al., 2014, Def. 1.2)

$$\operatorname{aff}(\mathcal{S}_k, \mathcal{S}_\ell) := \frac{1}{\sqrt{\min\{d_k, d_\ell\}}} \left\| \mathbf{U}^{(k)\top} \mathbf{U}^{(\ell)} \right\|_F \tag{2.6}$$

and can equivalently be expressed in terms of the principal angles $\theta_1 \leq \ldots \leq \theta_{\min\{d_k, d_\ell\}}$ between $\mathcal{S}_k$ and $\mathcal{S}_\ell$ (Golub and Van Loan, 1996, Sec. 6.3.4) according to

$$\operatorname{aff}(\mathcal{S}_k, \mathcal{S}_\ell) = \sqrt{\frac{\cos^2(\theta_1) + \ldots + \cos^2(\theta_{\min\{d_k, d_\ell\}})}{\min\{d_k, d_\ell\}}}. \tag{2.7}$$

We have $0 \leq \operatorname{aff}(\mathcal{S}_k, \mathcal{S}_\ell) \leq 1$ and for subspaces intersecting in t dimensions, we get $\cos(\theta_1) = \ldots = \cos(\theta_t) = 1$ and hence $\operatorname{aff}(\mathcal{S}_k, \mathcal{S}_\ell) \geq \sqrt{t/\min\{d_k, d_\ell\}}$.

Recall that spectral clustering recovers the oracle segmentation

$\{\mathcal{Y}_1, \ldots, \mathcal{Y}_L\}$ if $\hat{L} = L$ and each connected component in G corresponds to one of the $\mathcal{Y}_\ell$. Establishing conditions that guarantee zero clustering error is inherently difficult. To the best of our knowledge the only instances of such a result for spectral clustering-based subspace clustering algorithms are (Heckel and Bölcskei, 2015, Thm. 2) for TSC in the noiseless case and a condition in (Wang et al., 2016b) guaranteeing that a post-processing procedure for SSC yields correct clustering in the noisy case. We will rely on the following intermediate, albeit sensible, performance measure, which has become standard in the subspace clustering literature and was also employed in (Soltanolkotabi and Candès, 2012; Soltanolkotabi et al., 2014; Dyer et al., 2013; Heckel and Bölcskei, 2015; Wang and Xu, 2016; Heckel et al., 2017; Dyer et al., 2013; You et al., 2016; Park et al., 2014).

Definition 2.1 (No false connections (NFC) property). *The graph G satisfies the no false connections (NFC) property if, for all $\ell \in [L]$, the nodes corresponding to $\mathcal{Y}_\ell$ are connected to other nodes corresponding to $\mathcal{Y}_\ell$ only.*

In what follows, we often say "SSC-OMP/SSC-MP satisfies the NFC property" instead of "the graph G generated by SSC-OMP/SSC-MP satisfies the NFC property". To guarantee perfect clustering, we would need to ensure—in addition to the NFC property—that the subgraphs of G corresponding to the $\mathcal{Y}_\ell$ are connected. This would preclude split-ups of the subgraphs of G corresponding to the individual $\mathcal{Y}_\ell$. Sufficient conditions guaranteeing this property for SSC were established in (Nasihatkon and Hartley, 2011) for $m = 3$ in the noiseless case.

Note that the NFC property does not involve the parameter $\hat{L}$. The sufficient conditions for SSC-OMP and SSC-MP to satisfy the NFC property reported next, therefore, do not require $\hat{L} = L$.

Our main result for SSC-OMP with DI-stopping is the following.

Theorem 2.1 (SSC-OMP with DI-stopping). *Define the sampling density $\rho_\ell := (n_\ell - 1)/d_\ell$, and let $d_{\max} := \max_{\ell \in [L]} d_\ell$ and $\rho_{\min} = \min_{\ell \in [L]} \rho_\ell$. Assume that $m \geq 2d_{\max}$, $\rho_{\min} \geq c_\rho$, $\sigma \leq 1/2$, and $s_{\max} \leq$*

$\min_{\ell\in[L]}\{c_s d_\ell/\log((n_\ell-1)e/s_{\max})\}$, *where* c_ρ *and* c_s *are numerical constants satisfying* $c_\rho > 1$, $0 < c_s \leq 1/10$. *Then, the clustering condition*

$$\max_{k,\ell\colon k\neq\ell} \operatorname{aff}(\mathcal{S}_k,\mathcal{S}_\ell) + \frac{10\sigma}{\sqrt{\log(N^3 s_{\max})}}\left(\frac{\sqrt{d_{\max}}}{\sqrt{m}}c(\sigma) + \frac{\sqrt{2}}{\sqrt{\rho_{\min}}}\left(1+\frac{3}{2}\sigma\right)\right) \leq \frac{1}{8\log(N^3 s_{\max})} \qquad (2.8)$$

with $c(\sigma) = 10 + 13\sigma$ *guarantees that the graph* G *generated by SSC-OMP under DI-stopping satisfies the NFC property w.p. at least*

$$P^\star := 1 - 6/N - 5Ne^{-c_m m} - 6\sum_{\ell\in[L]} n_\ell e^{-c_d d_\ell} \qquad (2.9)$$

for numerical constants c_d *and* c_m *obeying* $0 < c_d \leq 1/18$, $0 < c_m \leq 1/8$.

The main result for SSC-MP with DI-stopping is as follows.

Theorem 2.2 (SSC-MP with DI-stopping). *Define the sampling density* $\rho_\ell := (n_\ell - 1)/d_\ell$, *and let* $d_{\max} := \max_{\ell\in[L]} d_\ell$ *and* $\rho_{\min} = \min_{\ell\in[L]}\rho_\ell$. *Assume that* $m \geq 2d_{\max}$, $\rho_{\min} \geq c_\rho$, $\sigma \leq 1/2$, $s_{\max} > 0$, *and* $p_{\max} \leq \min_{\ell\in[L]}\{c_s d_\ell/\log((n_\ell-1)e/p_{\max})\}$, *where* c_ρ *and* c_s *are numerical constants satisfying* $c_\rho > 1$, $0 < c_s \leq 1/10$. *Then, the clustering condition* (2.8) *with* $c(\sigma) = 22 + 29\sigma$ *guarantees that the graph* G *generated by SSC-MP under DI-stopping satisfies the NFC property w.p. at least* $P^\star$ *as defined in* (2.9).

The proofs of Theorems 2.1 and 2.2 can be found in Appendices 2.A and 2.B, respectively.

Theorems 2.1 and 2.2 essentially state that SSC-OMP and SSC-MP satisfy the NFC property for $s_{\max}$ and $p_{\max}$ linear—up to log-terms—in $d_{\min} := \min_{\ell\in[L]} d_\ell$, provided that the subspaces are not too close (in terms of their pairwise affinities), the noise variance σ^2 is sufficiently small, and the data set $\mathcal{Y}$ contains sufficiently many points from each subspace $\mathcal{S}_\ell$. Specifically, the clustering condition (2.8) tells us that the subspaces $\mathcal{S}_\ell$ are allowed to be quite close to each other and can even intersect in a substantial fraction of their dimensions, all provided that σ^2 is not too large. Moreover, inspection of the second term on the left-hand side (LHS) of (2.8) shows that a higher noise variance σ^2 is tolerated when m becomes large relative to the largest subspace dimension $d_{\max}$ and/or the data set $\mathcal{Y}$ contains an increasing number of points in each of the subspaces, resulting in an increase in the minimum sampling density $\rho_{\min}$. The clustering condition (2.8) can hence be satisfied under the condition $\sigma \leq 1/2$ imposed by Theorems 2.1 and 2.2 if m is sufficiently large relative to $d_{\max}$ and if $\rho_{\min}$ is sufficiently large (but not too large, in order to prevent the right-hand side (RHS) of (2.8) from becoming too small; for example, $\rho_{\min}$ should not scale exponentially in one of the d_ℓ). This shows that SSC-OMP and SSC-MP, indeed, satisfy the NFC property even when the noise variance σ^2 is on the order of the signal energy, i.e., when the signal to noise ratio $\mathrm{SNR} := \mathbb{E}[\|\mathbf{x}_j\|_2^2] / \mathbb{E}[\|\mathbf{z}_j\|_2^2] = 1/\sigma^2$ satisfies $\mathrm{SNR} \approx 0\mathrm{dB}$ (recall that $\mathbf{y}_i^{(\ell)} = \mathbf{x}_i^{(\ell)} + \mathbf{z}_i^{(\ell)}$ with $\mathbb{E}[\|\mathbf{x}_i^{(\ell)}\|_2^2] = 1$).

The RHS of (2.8) going to zero as $N \to \infty$ may appear counter-intuitive as one would expect clustering to become easier when the number of data points increases. Note, however, that (2.8) allows the subspaces to intersect, and Theorems 2.1 and 2.2 guarantee the NFC property for *all data points*. Now, when N increases, owing to the statistical data model our analysis is based on, the number of data points that are close to the intersection of two subspaces also increases, which in turn leads to an increase in the probability of the NFC property being violated for at least one data point. This then results in the clustering condition becoming more restrictive. The clustering conditions for SSC in (Soltanolkotabi et al., 2014, Eq. (3.1)), (Wang and Xu, 2016, Thm. 10) and for TSC in (Heckel and Bölcskei, 2015, Eq. (8)) exhibit the same $O(1/\log(N))$ scaling and hence the same seemingly counter-intuitive behavior.

We hasten to add that the condition $\sigma \leq 1/2$ in Theorems 2.1 and 2.2 was imposed only to get clustering conditions that are of simple form. Removing the restriction $\sigma \leq 1/2$ (which is used to get the bounds (2.28) and (2.29) in Appendix 2.A) would lead to clustering conditions allowing, in principle, for arbitrarily large σ (i.e., even for SNR $<$ 0dB), provided that the d_ℓ are sufficiently small compared to m, and $\rho_{\min}$ is sufficiently large. One might further expect that the upper bounds on $s_{\max}$ and $p_{\max}$ in Theorems 2.1 and 2.2, respectively, should depend on σ because the number of iterations for which SSC-OMP and SSC-MP are guaranteed to select points from $\mathcal{Y}_\ell \backslash \{\mathbf{y}_j\}$ for $\mathbf{y}_j \in \mathcal{Y}_\ell$ should decrease as σ increases. However, this is not the case as the clustering condition (2.8) limits the noise variance (more precisely, the variance of the noise components on the subspaces) depending on $\max_{k,\ell\colon k\neq\ell} \operatorname{aff}(\mathcal{S}_k, \mathcal{S}_\ell)$, N, $\rho_{\min}$, and $s_{\max}$. We furthermore note that the conditions in Theorems 2.1 and 2.2 (with different constants in (2.8)) continue to guarantee the NFC property for bounded noise or sub-gaussian noise, in both cases of isotropic distribution. It is interesting to see that SSC-MP satisfies the NFC property under virtually the same conditions as SSC-OMP, although in practice SSC-MP typically exhibits a lower running time at fixed performance.

Comparing the clustering condition (2.8) to those for SSC in (Soltanolkotabi et al., 2014, Thm. 3.1) and for TSC in (Heckel and Bölcskei, 2015, Thm. 3), both of which guarantee the NFC property and apply to the same data model as used here, we find that (2.8) exhibits the same structure (up to log-factors and constants) apart from the term proportional to $\sqrt{1/\rho_{\min}}$ on the LHS of (2.8). This term dominates the term proportional to $\sqrt{d_{\max}/m}$ only if $\sqrt{d_{\max}/m} \ll \sqrt{1/\rho_{\min}} = \sqrt{\max_{\ell\in[L]}(d_\ell/(n_\ell - 1))}$, i.e., if $\max_{\ell\in[L]} n_\ell \ll m$ (owing to $\max_{\ell\in[L]}(d_\ell/(n_\ell - 1)) \geq (\max_{\ell\in[L]} d_\ell)/(\max_{\ell\in[L]} n_\ell - 1) > d_{\max}/(\max_{\ell\in[L]} n_\ell)$). Similarly, the clustering condition in (Wang and Xu, 2016, Thm. 10) does not have a term proportional to $\sqrt{1/\rho_{\min}}$ as (2.8) does, but imposes a slightly more restrictive condition on σ, requiring $\sigma(c+\sigma)$ to be at most on the order of $\sqrt{m-d}/d$ instead of $\sqrt{m}/\sqrt{d}$ (assuming $d_\ell = d$ for all $\ell \in [L]$ and neglecting log-terms for simplicity of exposition), where c is a constant. Numerical results in Section 2.3.1 indicate that the term proportional to $\sqrt{1/\rho_{\min}}$ in (2.8) is not an artifact of our proof techniques, but rather fundamental.

We further note that setting $\sigma = 0$, the second term on the LHS of (2.8) vanishes and we recover (up to log-factors and constants) the clustering condition

$$\max_{k,\ell:\, k \neq \ell} \operatorname{aff}(\mathcal{S}_k, \mathcal{S}_\ell) \leq \frac{\sqrt{\log(\rho_{\min})}}{64 \log(N)}$$

found in (Heckel et al., 2017, Cor. 1) for SSC-OMP in the noiseless case.

In summary, SSC-OMP, SSC-MP, TSC, and SSC all satisfy the NFC property under similar (sufficient) conditions, while differing considerably w.r.t. computational complexity. Specifically, SSC-OMP and SSC-MP, albeit greedy, are computationally more expensive than TSC, but significantly less expensive than SSC. On the other hand, SSC-MP can outperform TSC quite significantly in certain applications (see Section 2.3.2). A detailed comparison of SSC, SSC-OMP, SSC-MP, and TSC in terms of performance and running times is provided in Section 2.3.2. The performance of all four algorithms varies across data sets, and none of the algorithms consistently outperforms the other ones.

Recall that under DI-stopping the choice of the parameters $s_{\max}$, $p_{\max}$ is critical for the success of SSC-OMP and SSC-MP. Taking $s_{\max}, p_{\max}$ too small or too large may lead to cluster split-ups or to many false connections, respectively. The maximum range for $s_{\max}$, $p_{\max}$ for our results to guarantee the NFC property is determined (up to log-factors) by the smallest subspace dimension $d_{\min}$, which is usually unknown. Furthermore, if $d_{\min}$ is small, the range of admissible values for $s_{\max}$, $p_{\max}$ will also be small. The clustering condition (2.8) is, however, only sufficient (for the NFC property to hold) and good clustering performance may be obtained in practice for larger values of $s_{\max}$, $p_{\max}$ than those identified by Theorems 2.1 and 2.2.

We proceed to our main result on DD-stopping, which indicates that the problems with choosing $s_{\max}$, $p_{\max}$ for DI-stopping due to unknown d_ℓ can be mitigated—to a certain extent—through DD-stopping. Specifically, we show that SSC-OMP and SSC-MP under DD-stopping automatically select at least on the order of d_ℓ points from $\mathcal{Y}_\ell$ to represent $\mathbf{y}_j \in \mathcal{Y}_\ell$. We hasten to add, however, that Theorem 2.3 does not guarantee that no additional data points corresponding to false connections are selected and Theorem 2.3 hence does not guarantee the NFC property.

Theorem 2.3 (SSC-OMP and SSC-MP with DD-stopping)**.** *Define the sampling density $\rho_\ell := (n_\ell - 1)/d_\ell$, and let $d_{\max} := \max_{\ell \in [L]} d_\ell$ and $\rho_{\min} = \min_{\ell \in [L]} \rho_\ell$. Suppose that $m \geq 2d_{\max}$, $\rho_{\min} \geq c_\rho$, and $\sigma \leq 1/2$, where c_ρ is a numerical constant satisfying $c_\rho > 1$. Pick $\tau \in [0, 2/3 - (\sqrt{d_{\max}}/\sqrt{m})\sigma]$. Then, the clustering condition* (2.8) *with $s_{\max}$ on both sides replaced by $\max_{\ell \in [L]} \lfloor c_s d_\ell / \log((n_\ell - 1)e) \rfloor$ guarantees w.p. at least $P^\star$ as defined in* (2.9)*, for all $\mathbf{y}_j \in \mathcal{Y}_\ell$, $j \in [n_\ell]$, $\ell \in [L]$, that the corresponding coefficient vectors $\mathbf{b}_j$ computed by OMP and MP (if it terminates) have at least*

$$\left\lfloor \frac{d_\ell}{\log((n_\ell - 1)e)} \min \left\{ \frac{1}{3} \left(\frac{2}{3} - \frac{\tau}{1 - \frac{3}{2} \frac{\sqrt{d_\ell}}{\sqrt{m}} \sigma} \right)^2, c_s \right\} \right\rfloor \tag{2.10}$$

non-zero entries corresponding to points in $\mathcal{Y}_\ell \backslash \{\mathbf{y}_j\}$.

Note that MP is not guaranteed to terminate under the conditions of Theorem 2.3 as τ in the admissible range indicated by Theorem 2.3 could be too small for termination (see the corresponding discussion in Section 2.1). The ensuing statements on SSC-MP all apply only if MP, indeed, terminates for all points $\mathbf{y}_j \in \mathcal{Y}$. Theorem 2.3 identifies a range for the threshold parameter τ guaranteeing that both SSC-OMP and SSC-MP deliver a graph G which has each $\mathbf{y}_j \in \mathcal{Y}_\ell$, $\ell \in [L]$, connected to at least $O(d_\ell / \log(n_\ell - 1))$ other points in $\mathcal{Y}_\ell$. If σ increases, the probability of OMP and MP selecting false connections increases and more iterations need to be performed for a given number of true connections to be selected. As a consequence, the interval for τ specified in Theorem 2.3 decreases as σ increases. As already pointed out, Theorem 2.3 does not guarantee the NFC property, and choosing τ too small will result in entries in the coefficient vectors $\mathbf{b}_j$ that correspond to false connections. Intuitively, we expect that choosing τ sufficiently large, OMP and MP should stop early enough so as to avoid false connections. More specifically, one would expect that τ needs to be chosen larger as σ increases so as to avoid OMP and MP selecting points from $\mathcal{Y} \backslash \mathcal{Y}_\ell$ to represent $\mathbf{y}_j \in \mathcal{Y}_\ell$. Unfortunately, it seems rather difficult, at least for the statistical data model at hand, to analytically characterize a range for τ that guarantees the NFC property and simultaneously on the order of d_ℓ connections between $\mathbf{y}_j \in \mathcal{Y}_\ell$ and other points in $\mathcal{Y}_\ell$, for all $j \in [n_\ell]$, $\ell \in [L]$. Nonetheless, it turns out, that in practice G often exhibits both of these properties if τ is chosen appropriately. Numerical results in Section 2.3.3 corroborate this claim. In summary, SSC-OMP and SSC-MP under DD-stopping with appropriately chosen τ detect the dimensions of the subspaces $\mathcal{S}_\ell$ correctly and adapt the sparsity level of the individual representations according to the dimension of the subspace the data point at hand lies in.

The procedure in (Soltanolkotabi et al., 2014, Alg. 2) for the selection of a per-data-point Lasso parameter in SSC has a similar subspace dimension-detecting property but comes with stronger theoretical guarantees. Specifically, (Soltanolkotabi et al., 2014, Thm. 3.1) and (Soltanolkotabi et al., 2014, Thm. 3.2) taken together guarantee the NFC property and, concurrently, that each $\mathbf{y}_j \in \mathcal{Y}_\ell$ is connected to on the order of d_ℓ other points in $\mathcal{Y}_\ell$, all this provided that the noise variance—assumed known—is small enough. The procedure in (Soltanolkotabi et al., 2014, Alg. 2) could also be employed to select $s_{\max}$ in SSC-OMP and SSC-MP under DI-stopping for each data point individually. This emulates DD-stopping by employing DI-stopping together with a data-point-wise parameter selection procedure. More specifically, as shown in (Soltanolkotabi et al., 2014, Lem. A.2), the optimal cost of the auxiliary Lasso problem in (Soltanolkotabi et al., 2014, Eq. (2.4)) for $\mathbf{y}_j \in \mathcal{Y}_\ell$ is proportional to $\sqrt{d_\ell}$. Squaring the optimal cost therefore yields an estimate of d_ℓ, which can, in turn, be used to select the parameter $s_{\max}$ in SSC-OMP such that it is on the order of d_ℓ for $\mathbf{y}_j \in \mathcal{Y}_\ell$ and thereby satisfies the condition in Theorem 2.1 (we can lower-bound the factor $1/\log((n_\ell - 1)e/s_{\max})$ in that condition by $1/\log(Ne)$). This would then guarantee, in addition to the NFC property, on the order of $d_\ell/\log(n_\ell)$ true connections for $\mathbf{y}_j \in \mathcal{Y}_\ell$, $j \in [n_\ell]$, $\ell \in [L]$, for both SSC-OMP and SSC-MP under the conditions of Theorems 1 and 2, and would hence realize the "many true discoveries" and the NFC property at the same time as guaranteed by (Soltanolkotabi et al., 2014, Thm. 3.1) and (Soltanolkotabi et al., 2014, Thm. 3.2) for SSC. We point out, however, that the selection procedure in (Soltanolkotabi et al., 2014, Alg. 2) results in considerable computational burden in addition to solving the (already computationally demanding) N Lasso problems required by SSC or running the OMP and MP routines in SSC-OMP and SSC-MP, respectively, to perform the actual clustering.

Finally, we note that determining the range of admissible threshold parameters $\tau \in [0, 2/3 - (\sqrt{d_{\max}}/\sqrt{m})\sigma]$ in Theorem 2.3 requires knowledge of the noise variance σ^2. In principle, knowledge of $d_{\max}$ is required as well. We can, however, upper-bound $\sqrt{d_{\max}}/\sqrt{m}$ by 1 thereby obviating the need for knowing $d_{\max}$ at the cost of a reduced range for τ.

2.3. NUMERICAL RESULTS[2]

We compare the performance of SSC-OMP, SSC-MP, SSC, TSC, and NSN. SSC-OMP and SSC-MP were implemented in Matlab exactly following their descriptions in Section 2.1. For SSC, TSC, and NSN, we used the implementations provided in the corresponding references (Elhamifar and Vidal, 2013), (Heckel and Bölcskei, 2015), and (Park et al., 2014), respectively.

Our main performance measure is the clustering error (CE), i.e., the fraction of misclustered data points, defined as

$$\mathrm{CE}(\hat{\mathbf{c}}, \mathbf{c}) = \min_{\pi} \left(1 - \frac{1}{N} \sum_{i=1}^{N} 1_{\{\pi([\hat{\mathbf{c}}]_i) = [\mathbf{c}]_i\}} \right), \tag{2.11}$$

[2] Code to reproduce the experiments is available at `http://www.nari.ee.ethz.ch/commth/research/`.

where $\mathbf{c} \in [L]^N$ and $\hat{\mathbf{c}} \in [L]^N$ are the true and the estimated assignments, respectively, and the minimum is taken over all permutations $\pi\colon [L] \to [L]$. Additional performance measures will be introduced in the description of the respective experiments.

We provide the algorithms with the true number of subspaces L in all experiments. All running times were measured on a PC with 32 GB RAM and 4-core Intel Core i7-3770K CPU clocked at 3.50 GHz. It is quite common in the computer vision literature to perform post-processing on the adjacency matrix $\mathbf{A}$ generated by the individual clustering algorithms. This can improve the clustering performance, but will not be pursued here in order to simplify our comparisons.

2.3.1. Comparison of SSC-OMP and SSC-MP

As the clustering condition (2.8) is only a sufficient condition and guarantees the NFC property only, it is unclear to what extent the behavior predicted by (2.8) is reflected in the CE. The following experiment is devoted to answering this question while comparing SSC-OMP and SSC-MP. We generate data sets $\mathcal{Y}$ according to the statistical data model described in Section 2.2. Specifically, we set $d_\ell = d$, $\ell \in [L]$, and we choose the bases $\mathbf{U}^{(\ell)} \in \mathbb{R}^{m \times d}$ to all intersect in a shared t-dimensional space and to be mutually orthogonal on the orthogonal complement of this intersection. More specifically, the bases $\mathbf{U}^{(\ell)}$ are obtained by choosing a matrix $\mathbf{U} \in \mathbb{R}^{m \times (L(d-t)+t)}$ uniformly at random from the set of all orthonormal matrices of dimension $m \times (L(d-t)+t)$ and setting $\mathbf{U}^{(\ell)} := [\mathbf{U}_{[t]}\ \mathbf{U}_{\mathcal{T}_\ell}]$, where $\mathcal{T}_\ell := \{t+(\ell-1)(d-t)+1, \dots, t+\ell(d-t)\}$. This results in $\operatorname{aff}(\mathcal{S}_k, \mathcal{S}_\ell) = \sqrt{t/d}$, $k, \ell \in [L]$, $k \neq \ell$. Varying the parameter t therefore allows us to vary the pairwise affinities. We furthermore set $n_\ell = n$, $\ell \in [L]$, and generate instances of $\mathcal{Y}$ by sampling n data points uniformly at random from each subspace and adding $\mathcal{N}(\mathbf{0}, (\sigma^2/m)\mathbf{I})$ noise to each data point (the noise vectors are generated independently across data points). We let $L = 3$, $d = 20$, $m = 200$, and vary t, $\rho_{\min} = \rho = n/d$, and σ^2. Furthermore, we employ DI-stopping and set $s_{\max} = d/2 = 10$ for both SSC-OMP and SSC-MP, and $p_{\max} = N$. Figure 2.2 shows the CE as a function of $\max_{k,\ell\colon k\neq\ell} \operatorname{aff}(\mathcal{S}_k,\mathcal{S}_\ell) = \sqrt{t/d}$, ρ, and σ^2. The results nicely reflect the qualitative behavior indicated by the clustering condition (2.8). Specifically, both SSC-OMP and SSC-MP tolerate higher noise variance as the affinities between the subspaces decrease and the number of points in $\mathcal{Y}$ drawn from each subspace, n, increases. It is furthermore interesting to observe that the performance of SSC-OMP and SSC-MP is virtually identical.

Recall that the clustering condition (2.8), apart from the term proportional to $\sqrt{1/\rho_{\min}}$, exhibits the same scaling behavior as those guaranteeing the NFC property for SSC in (Soltanolkotabi et al., 2014, Thm. 3.1) and for TSC in (Heckel and Bölcskei, 2015, Thm. 3). To find out whether this additional term is an artifact of our proof technique, we first note that the clustering condition (2.8) takes the form

$$\max_{k,\ell\colon k\neq\ell} \operatorname{aff}(\mathcal{S}_k,\mathcal{S}_\ell) + \frac{c_1}{\sqrt{\rho}} \leq c_2, \tag{2.12}$$

for fixed σ, and

$$\sigma(c_3 + \sigma c_4 + \frac{1}{\sqrt{\rho}}(c_5 + \sigma c_6)) \leq c_7, \tag{2.13}$$

for fixed maximum affinity, where c_1-$c_7 > 0$ are constants, $d_{\max}$ and m were assumed constant in both cases, and factors logarithmic in any of the parameters (variable or fixed) were neglected. Rewriting (2.12) and (2.13) assuming equality, we get

$$\rho = \left(\frac{c_1}{c_2 - \max_{k,\ell\colon k\neq\ell} \operatorname{aff}(\mathcal{S}_k,\mathcal{S}_\ell)}\right)^2 \tag{2.14}$$

and

$$\rho = \left(\frac{\sigma(c_5 + c_6\sigma)}{c_7 - \sigma(c_3 + \sigma c_4)}\right)^2, \tag{2.15}$$

respectively. In the top and bottom rows of Figure 2.2, we now fit (2.14) and (2.15), respectively (by manually adjusting the constants c_1-c_7), to the boundaries between the regions of success and the regions of failure. The shape of the fitted curves follows the boundaries indicated

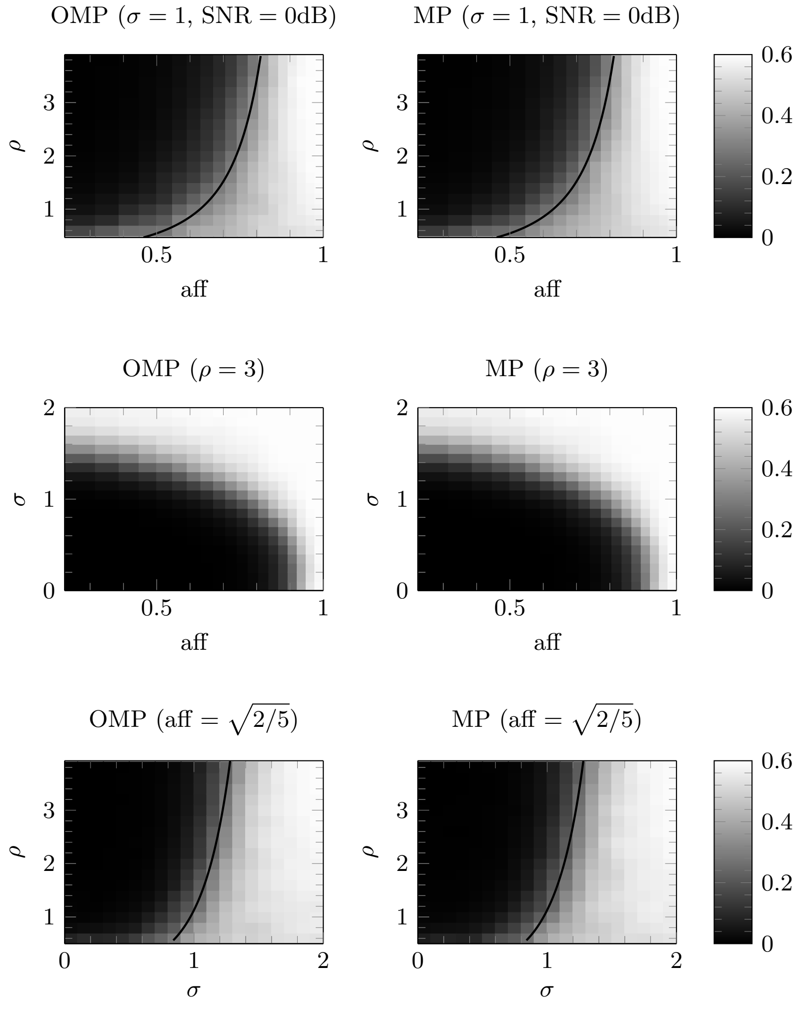


Fig. 2.2: CE as a function of $\mathrm{aff} := \max_{k,\ell\colon k\neq\ell} \mathrm{aff}(\mathcal{S}_k, \mathcal{S}_\ell)$, $\rho = n/d$, and σ^2. The (fitted) black curves in the top and bottom rows correspond to the curves (2.14) and (2.15), respectively, and delineate the boundary between success and failure.

by the numerical results closely. This can be taken as an indication—at least to a certain extent—of the term in (2.8) proportional to $\sqrt{1/\rho_{\min}}$ being fundamental.

2.3.2. Face clustering

Next, we consider the problem of clustering face images of different individuals taken under varying illumination conditions. The rationale for employing subspace clustering to solve this problem stems from the observation that vectorized images of a given individual taken under different lighting conditions lie (approximately) in a 9-dimensional linear subspace (Basri and Jacobs, 2003). We consider the Extended Yale B data set (Georghiades et al., 2001; Lee et al., 2005), which contains 192 × 168 pixel frontal face images of 38 individuals, each taken under 64 different illumination conditions. As SSC-MP does not seem to have been considered before in the literature, we compare its performance in terms of CE to that of SSC-OMP, SSC, TSC, and NSN. Similar experiments comparing the performance of SSC-OMP to that of other subspace clustering algorithms for face clustering were presented in (Dyer et al., 2013; You et al., 2016; Heckel et al., 2017). To accomodate the memory requirements incurred by the optimization problems in SSC, we apply SSC (and, to ensure a fair comparison, also the other algorithms) to the downsampled 48 × 42 pixel versions of the images in the Extended Yale B data set provided in (Elhamifar and Vidal, 2013). Note, however, that SSC-OMP, SSC-MP, TSC, and NSN all could handle the original 192×168 pixel images. Instances of the data set $\mathcal{Y}$ are obtained by first choosing a subset of L individuals uniformly at random from the set of all individuals and then collecting the 64 (vectorized) images corresponding to each of the chosen individuals. As DD-stopping leads to a large number of false connections—arguably due to (sparse) corruptions induced by shadows and specular reflections in the face images (Elhamifar and Vidal, 2013)—we rely on DI-stopping (i.e., $\tau = 0$) with $s_{\max} = 5$ (which corresponds to the choice made in (Dyer et al., 2013) for SSC-OMP) for both SSC-OMP and SSC-MP, and we set $p_{\max} = N$.

Note that both OMP and MP perform exactly $s_{\max}$ iterations in this experiment as the points in $\mathcal{Y}$ are in general position. For SSC, TSC, and NSN we use the parameter values employed for the face clustering experiments in (Elhamifar and Vidal, 2013), (Heckel and Bölcskei, 2015), and (Park et al., 2014), respectively. We emphasize that the experiments in (Elhamifar and Vidal, 2013), (Heckel and Bölcskei, 2015), and (Park et al., 2014) all also provide the true number of subspaces L to the individual algorithms. Note furthermore that we employ SSC with the objective function as formulated in (Elhamifar and Vidal, 2013) accounting for sparse corruptions of the data points.

Table 2.1 shows the CE for different choices of L, averaged over 100 instances of $\mathcal{Y}$ for each L. In Table 2.2, we report the corresponding average running times. SSC-MP outperforms SSC-OMP (and TSC) for all values of L, but $L = 2$, and does so at consistently lower running times (recall that SSC-OMP and SSC-MP both perform exactly $s_{\max}$ iterations in this experiment, but SSC-MP has a lower per-iteration cost than SSC-OMP). Furthermore, SSC-OMP uniformly outperforms SSC. The lowest CE is obtained with NSN. We note, however, that the performance of SSC-MP, except for $L = 2$, almost matches that of NSN, and SSC-MP consistently has roughly half the running time of NSN. TSC has the lowest running time but yields the largest CE (which is particularly high for this very data set (Heckel and Bölcskei, 2015)). The running time of SSC is one to two orders of magnitude higher than that of all other algorithms. Also note that the difference between the running times of SSC-OMP and SSC-MP is small because $s_{\max}$ is small. Finally, the difference between the results for SSC-OMP, SSC, TSC, and NSN reported here compared to the results reported in published works (You et al., 2016; Elhamifar and Vidal, 2013; Heckel and Bölcskei, 2015; Park et al., 2014) can be attributed to the fact that we do not perform post-processing on the adjacency matrix $\mathbf{A}$ produced by the individual algorithms.

Finally, we emphasize that for other applications such as, e.g., handwritten digit clustering (Heckel and Bölcskei, 2015), SSC and TSC yield lower CE than the other algorithms considered here, and none of the algorithms outperforms the others uniformly across applications as seen from experiments in (Elhamifar and Vidal, 2013; Heckel and Bölcskei, 2015; Park et al., 2014).

Table 2.1: Average CE (in percent) for face clustering.

L	2	3	5	8	10
SSC-OMP	2.83	4.04	6.81	12.98	14.14
SSC-MP	4.16	3.72	5.24	9.09	11.36
SSC	2.88	4.59	8.65	16.95	21.28
TSC	10.71	16.81	29.73	38.34	41.22
NSN	1.81	2.89	5.37	8.15	10.05

Table 2.2: Average running times (in seconds) for face clustering.

L	2	3	5	8	10
SSC-OMP	0.21	0.36	0.80	2.01	3.13
SSC-MP	0.15	0.28	0.65	1.74	2.80
SSC	12.63	17.50	28.24	45.78	60.33
TSC	0.08	0.14	0.29	0.62	0.95
NSN	0.34	0.55	1.10	3.05	5.64

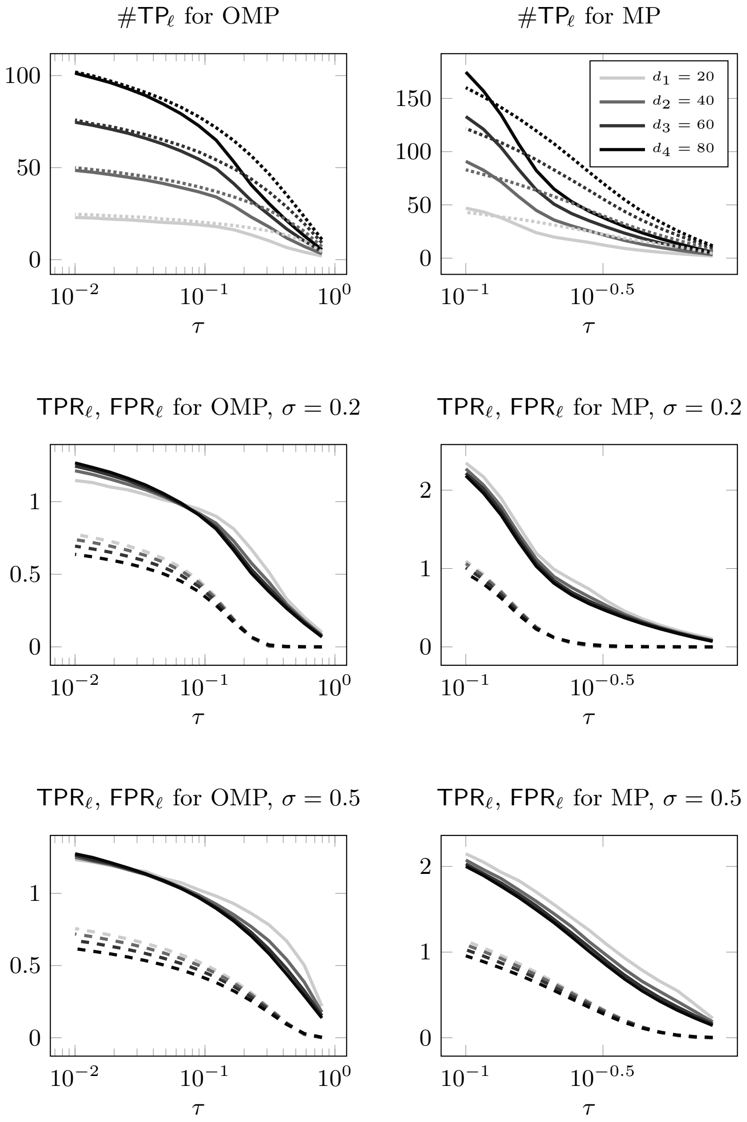
2.3.3. True positives/false positives tradeoff in DD-stopping

Recall that for DD-stopping Theorem 2.3 guarantees that each data point $\mathbf{y}_j \in \mathcal{Y}_\ell$, $\ell \in [L]$, is connected (in G) to at least $O(d_\ell / \log(n_\ell - 1))$ other points in $\mathcal{Y}_\ell$; we here refer to such connections as true positives (TP). As already mentioned, Theorem 2.3, does, however, not guarantee the absence of false connections and provides a lower bound on the number of TP only. It is therefore not clear, a priori, to what extent SSC-OMP and SSC-MP under DD-stopping, indeed, do detect the subspace dimensions. First, recall that by "detecting the subspace dimensions" we mean that the coefficient vectors $\mathbf{b}_j$ for all $\mathbf{y}_j \in \mathcal{Y}_\ell$, $\ell \in [L]$, have on the order of d_ℓ non-zero entries corresponding to TP and essentially no non-zero entries corresponding to false connections. We designate the number of TP as $\#\mathsf{TP}_\ell$, and the number of false positives (FP) as $\#\mathsf{FP}_\ell$, both averaged over the data points corresponding to the subspace $\mathcal{S}_\ell$. We will also need the notions of true positive rate (TPR) and false positive rate (FPR) defined, in this experiment, as $\mathsf{TPR}_\ell = \#\mathsf{TP}_\ell / d_\ell$ and $\mathsf{FPR}_\ell = \#\mathsf{FP}_\ell/(m - d_\ell)$, respectively.

We set $m = 300$, $L = 4$, $\rho_\ell = 4$, $\ell \in [L]$, and choose the basis matrices $\mathbf{U}^{(\ell)}$ for the subspaces of dimensions 20, 40, 60, and 80, according to $\mathbf{U}^{(\ell)} := [\bar{\mathbf{U}} \quad \tilde{\mathbf{U}}^{(\ell)}]$, where $\bar{\mathbf{U}}$ and the $\tilde{\mathbf{U}}^{(\ell)}$ are drawn uniformly at random from the set of all orthonormal matrices of dimensions 300×4 and $300 \times (d_\ell - 4)$, respectively. This guarantees that $\max_{k,\ell\colon k \neq \ell} \operatorname{aff}(\mathcal{S}_k, \mathcal{S}_\ell) \geq \sqrt{1/5}$ as the subspaces all intersect in a shared 4-dimensional space and possibly overlap in the orthogonal complement of this shared space. The data points are then drawn according to the statistical data model described in Section 2.2.

Figure 2.3 shows $\#\mathsf{TP}_\ell$ along with TPR_ℓ and FPR_ℓ as a function of τ and for different values of σ^2. The middle and bottom rows in Figure 2.3 show that the TPR curves corresponding to different ℓ are almost on top of each other, i.e., $\mathsf{TPR}_1(\tau) \approx \ldots \approx \mathsf{TPR}_4(\tau) \approx c_{\mathsf{TPR}}(\tau)$, which means that the number of TP for each subspace is, indeed, roughly proportional to the subspace dimension as $\#\mathsf{TP}_\ell(\tau) = \mathsf{TPR}_\ell(\tau) d_\ell \approx c_{\mathsf{TPR}}(\tau) d_\ell$, $\ell \in [L]$. This indicates that the result in Theorem 2.3 is order-wise optimal in d_ℓ. As, in addition, for large enough τ, $\mathsf{FPR}_\ell \approx 0$, $\ell \in [L]$, we conclude that SSC-OMP and SSC-MP, indeed, exhibit excellent subspace dimension detection properties provided that τ is chosen appropriately.

We finally note that a similar experiment investigating the TPR/FPR tradeoff as a function of the Lasso parameter in SSC was conducted in (Soltanolkotabi et al., 2014, Sec. 2.4.3) with the main conclusion that a Lasso parameter on the order of $1/\sqrt{d_\ell}$ extracts the subspace dimensions correctly order-wise with essentially no FP.



#TP$_\ell$ for OMP

#TP$_\ell$ for MP

TPR$_\ell$, FPR$_\ell$ for OMP, $\sigma = 0.2$

TPR$_\ell$, FPR$_\ell$ for MP, $\sigma = 0.2$

TPR$_\ell$, FPR$_\ell$ for OMP, $\sigma = 0.5$

TPR$_\ell$, FPR$_\ell$ for MP, $\sigma = 0.5$

Fig. 2.3: TP/FP tradeoff in DD-stopping as a function of τ. Top row: Solid lines: $\sigma = 0.2$, dotted lines: $\sigma = 0.5$. Middle and bottom rows: Solid lines: TPR, dashed lines: FPR.

2.3.4. Influence of $s_{\max}$ and $p_{\max}$ in DI-stopping

Recall that Theorems 2.1 and 2.2 provide a range of admissible values for $s_{\max}$ and $p_{\max}$ in DI-stopping. For small $d_{\min}$, these ranges are, however, small. As already pointed out, taking $s_{\max}$, $p_{\max}$ too small leads to cluster split-ups, whereas $s_{\max}$, $p_{\max}$ too large results in a large number of false connections in G. It is therefore important to determine the sensitivity of SSC-OMP and SSC-MP performance w.r.t. the choice of $s_{\max}$ and $p_{\max}$. The next experiment is devoted to this matter. We consider the CE as well as the average TPR and FPR which, in this experiment, are defined as $\mathsf{TPR}'_\ell = \#\mathsf{TP}_\ell/n_\ell$ and $\mathsf{FPR}'_\ell = \#\mathsf{FP}_\ell/(N-n_\ell)$, respectively. These alternative normalizations are used as we are not interested in investigating the dependence of TP and FP on the d_ℓ, as was done in Section 2.3.3; rather, we want to ensure that $\mathsf{TPR}'_\ell, \mathsf{FPR}'_\ell \in [0,1]$. Furthermore, we define the TP-ℓ_1-norm and the FP-ℓ_1-norm as the ℓ_1-norm of the entries of the coefficient vectors $\mathbf{b}_j$ corresponding to TP and FP, respectively, both averaged over all data points. The TP- and FP-ℓ_1-norms hence correspond to half of the average weight associated with TP and FP, respectively; the factor 1/2 stems from the adjacency matrix of G being given by $\mathbf{A} = \mathbf{B} + \mathbf{B}^\top$. The motivation for considering TP-/FP-ℓ_1-norms comes from the fact that the performance of spectral clustering is determined not only by the number of TP and FP, but also by the weights associated with the TP and the FP. Specifically, even when the $\mathbf{b}_j$ contain a considerable number of FP, good performance can still be obtained provided that the corresponding FP-ℓ_1-norm is sufficiently small.

We cluster synthetic data generated according to the statistical data model described in Section 2.2 as well as images taken from the Extended Yale B data set (we use downsampled 48 × 42 pixel versions of the images, see Section 2.3.2 for a detailed description). More specifically, in the case of synthetic data, we set $L = 3$, $m = 80$, $\sigma = 0.5$, $d_\ell = 15$, $\ell \in [L]$, and $\rho_\ell = 4$, $\ell \in [L]$, and we generate the bases $\mathbf{U}^{(\ell)}$, $\ell \in [L]$, to intersect in a shared 3-dimensional space by following the construction employed in Section 2.3.3 such that

$\max_{k,\ell\colon k\neq\ell} \operatorname{aff}(\mathcal{S}_k, \mathcal{S}_\ell) = \sqrt{1/5}$. In the case of face clustering, we follow the procedure described in Section 2.3.2 to obtain instances of $\mathcal{Y}$ containing the face images of $L = 3$ randomly selected individuals.

Figure 2.4 shows the CE, TPR/FPR, and TP-/FP-ℓ_1-norms as a function of $s_{\max}$ and $p_{\max}$. For both clustering problems the CE of SSC-OMP is seen to increase rapidly as a function of $s_{\max}$, while for SSC-MP with $p_{\max} = N$ (i.e., stopping is activated by $s = s_{\max}$), the CE increases very slowly for the face clustering problem and does not increase at all in the case of synthetic data. This indicates that SSC-MP exhibits significantly smaller sensitivity to the choice of $s_{\max}$ than SSC-OMP. As already pointed out in Section 2.1, this is due to the ability of SSC-MP to select points $\mathbf{y}_i \in \mathcal{Y}_\ell \backslash \{\mathbf{y}_j\}$ repeatedly to participate in the representation of $\mathbf{y}_j \in \mathcal{Y}_\ell$; for given $s_{\max}$ this results in SSC-MP producing fewer FP than SSC-OMP. Moreover, for SSC-MP, the TP-ℓ_1-norm exceeds the FP-ℓ_1-norm for all values of $s_{\max}$, while for SSC-OMP the FP-ℓ_1-norm exceeds the TP-ℓ_1-norm for large $s_{\max}$, and does so significantly. The coefficient vectors produced by SSC-MP hence lead to more favorable conditions for the spectral clustering step than those produced by SSC-OMP.

We further observe that the TPR and FPR of SSC-MP, with $s_{\max} = \infty$ (i.e., stopping is activated by $\|\mathbf{b}_j\|_0 = p_{\max}$), as a function of $p_{\max}$, increase at the same rate as the TPR and FPR of SSC-OMP as a function of $s_{\max}$. However, the ratio of TP- and FP-ℓ_1-norms for SSC-MP significantly exceeds that for SSC-OMP for all values of u (u is the variable on the x-axis in Figure 2.4 and corresponds to $s_{\max}$ for SSC-OMP and $p_{\max}$ for SSC-MP). In other words, even when we force the representations computed by SSC-MP and SSC-OMP to have the same sparsity level, thereby discounting the advantage MP has through its ability to reselect data points, SSC-MP still produces weight assignments in G that are more favorable in terms of spectral clustering. Indeed, in both the face clustering and the synthetic data clustering problem the CE incurred by SSC-OMP significantly exceeds that of SSC-MP for most values of u. In summary, this indicates that when we enforce the same target sparsity level for SSC-MP and SSC-OMP, SSC-MP is less sensitive to the choice of the sparsity level than

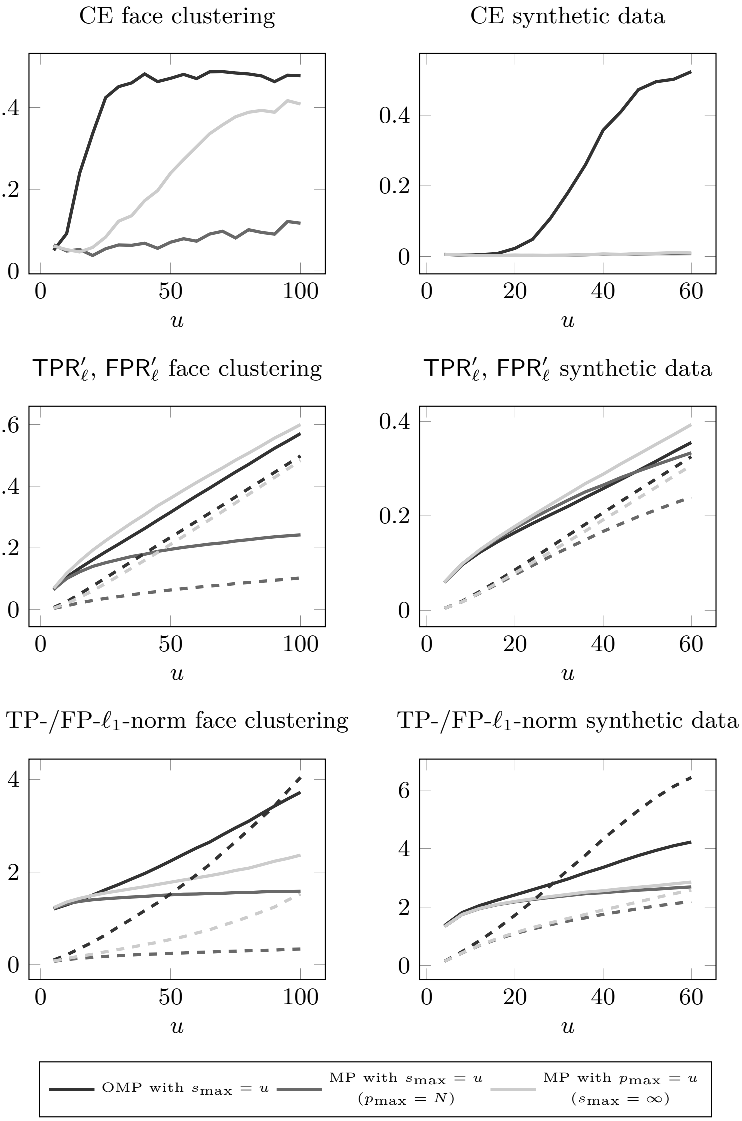


Fig. 2.4: Clustering performance of SSC-OMP and SSC-MP for DI-stopping, as a function of $s_{\max}$ and $p_{\max}$, respectively. Middle row: solid lines: TPR, dashed lines: FPR. Bottom row: solid lines: TP-ℓ_1-norm, dashed lines: FP-ℓ_1-norm.

SSC-OMP.

We finally note that while in the noisy case SSC-MP is much more robust than SSC-OMP w.r.t. the choice of the parameters for DI-stopping, in the noiseless case SSC-OMP yields slightly lower CE than SSC-MP for DI-stopping if the subspace affinities are large. This matter is investigated numerically in Appendix 2.3.5.

2.3.5. Influence of $s_{\max}$ and $p_{\max}$ in DI-stopping for noiseless data

We compare the influence of $s_{\max}$ and $p_{\max}$ on the performance of SSC-OMP and SSC-MP with DI-stopping and for noiseless data. To this end, we generate data lying in a union of three subspaces as described in Section 2.3.4 (with $\sigma = 0$), considering the pairs $(5, 3)$, $(10, 3)$, and $(10, 6)$ for (t, ρ), where t denotes the number of dimensions which the three subspaces intersect in. Figure 2.5 shows the CE along with the quantities TPR/FPR and the TP-/FP-ℓ_1-norm as a function of $s_{\max}$ (or $p_{\max}$ for SSC-MP if the maximum sparsity level is used as stopping criterion) in the range $\{1, \ldots, 2d\}$. We observe that SSC-OMP yields a slightly lower CE than SSC-MP for $t = 10$. While SSC-MP yields a higher TPR and a lower FPR than SSC-OMP for most of the values of t, ρ, and $s_{\max}$ or $p_{\max}$, SSC-MP assigns smaller values, than SSC-OMP, to entries in the adjacency matrix $\mathbf{A}$ corresponding to the true connections (see the plot in the last row, left, in Figure 2.5). This arguably leads to the slightly higher CE of SSC-MP compared to SSC-OMP for $t = 10$.

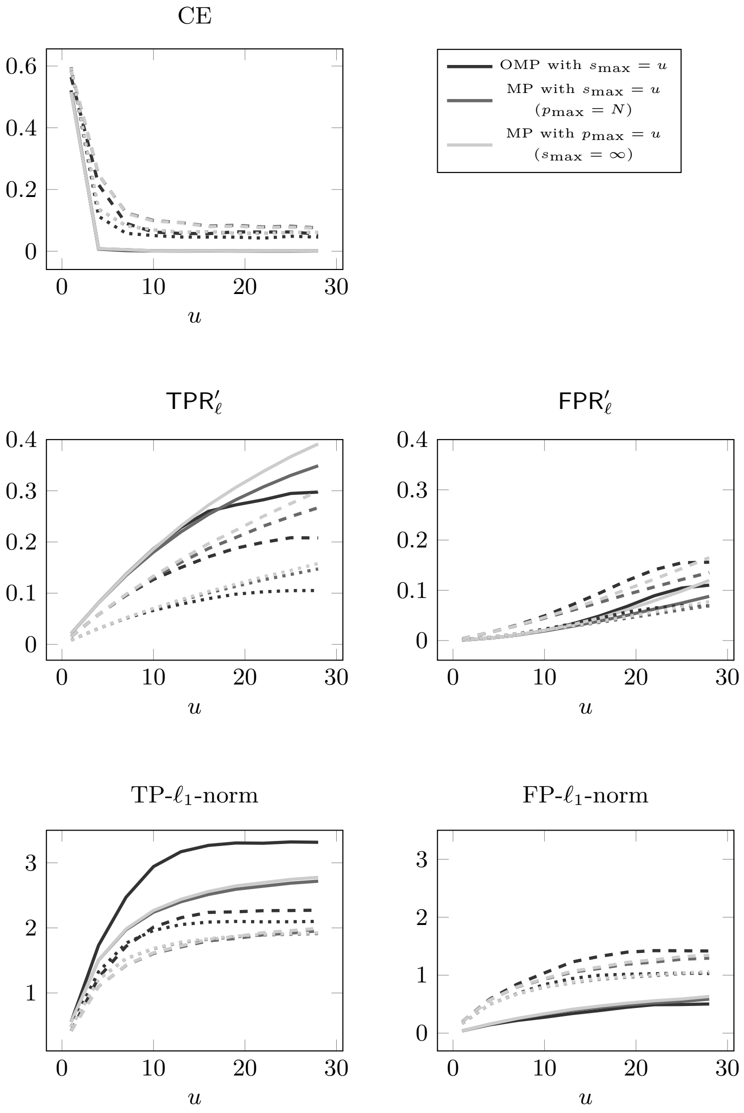


Fig. 2.5: Clustering performance of SSC-OMP and SSC-MP for DI-stopping, as a function of $s_{\max}$ and $p_{\max}$ for noiseless data. Solid line: $t = 5$, $\rho = 3$; dashed line: $t = 10$, $\rho = 3$; dotted line: $t = 10$, $\rho = 6$.

APPENDICES

2.A. PROOF OF THEOREM 2.1

Throughout the proof, we shall use $\mathbf{Y}^{(\ell)} := \mathbf{X}^{(\ell)} + \mathbf{Z}^{(\ell)} = \mathbf{U}^{(\ell)}\mathbf{A}^{(\ell)} + \mathbf{Z}^{(\ell)}$, $\mathbf{X}^{(\ell)} \in \mathbb{R}^{m \times n_\ell}$, $\mathbf{A}^{(\ell)} \in \mathbb{R}^{d_\ell \times n_\ell}$, and $\mathbf{Z}^{(\ell)} \in \mathbb{R}^{m \times n_\ell}$ to denote the matrices whose columns are the $\mathbf{y}_i^{(\ell)}$, $\mathbf{x}_i^{(\ell)}$, $\mathbf{a}_i^{(\ell)}$, and $\mathbf{z}_i^{(\ell)}$, $i \in [n_\ell]$, respectively. Furthermore, $\mathbf{P}_\| := \mathbf{U}^{(\ell)}\mathbf{U}^{(\ell)\top}$ and $\mathbf{P}_\perp := \mathbf{I} - \mathbf{U}^{(\ell)}\mathbf{U}^{(\ell)\top}$ stand for the orthogonal projection onto $\mathcal{S}_\ell$ and its orthogonal complement (in $\mathbb{R}^m$) $\mathcal{S}_\ell^\perp$, respectively. We do not indicate the dependence of $\mathbf{P}_\|$ and $\mathbf{P}_\perp$ on ℓ as this is always clear from the context. For $\mathbf{v} \in \mathbb{R}^m$ and $\mathbf{A} \in \mathbb{R}^{m \times n}$, we use the shorthands $\mathbf{v}_\| := \mathbf{P}_\|\mathbf{v}$, $\mathbf{A}_\| := \mathbf{P}_\|\mathbf{A}$, $\mathbf{v}_\perp := \mathbf{P}_\perp\mathbf{v}$, and $\mathbf{A}_\perp := \mathbf{P}_\perp\mathbf{A}$. Further, $\tilde{\mathbf{a}}_i^{(\ell)} \in \mathbb{R}^{d_\ell}$ denotes the coefficients of $\mathbf{y}_{i\|}^{(\ell)}$ in the basis $\mathbf{U}^{(\ell)}$, i.e., $\mathbf{y}_{i\|}^{(\ell)} = \mathbf{U}^{(\ell)}\tilde{\mathbf{a}}_i^{(\ell)}$, and similarly $\mathbf{Y}_\|^{(\ell)} = \mathbf{P}_\|\mathbf{Y}^{(\ell)} = \mathbf{U}^{(\ell)}\tilde{\mathbf{A}}^{(\ell)}$. Note that the distribution of $\tilde{\mathbf{a}}_i^{(\ell)} = \mathbf{a}_i^{(\ell)} + \mathbf{U}^{(\ell)\top}\mathbf{z}_i^{(\ell)}$ is rotationally invariant as $\mathbf{a}_i^{(\ell)}$ and $\mathbf{U}^{(\ell)\top}\mathbf{z}_i^{(\ell)}$ are statistically independent and both have rotationally invariant distributions. Finally, $\mathbf{r}_s(\mathbf{x}, \mathbf{D})$ and $\Lambda_s(\mathbf{x}, \mathbf{D})$ denote the residual and the index set, respectively, after iteration s, obtained by OMP applied to $\mathbf{x}$ with the columns of $\mathbf{D}$ as dictionary elements.

If $\min_{\ell \in [L]}\{c_s d_\ell / \log((n_\ell - 1)e/s_{\max})\} < 1$, then the condition in Theorem 2.1 admits zero OMP iterations, i.e., the graph G delivered by SSC-OMP has an empty edge set and thereby trivially no false connections. We therefore consider the case $1 \leq s_{\max} \leq \min_{\ell \in [L]}\{c_s d_\ell / \log((n_\ell - 1)e/s_{\max})\}$ in the remainder of the proof.

The graph G obtained by SSC-OMP has no false connections if for each $\mathbf{y}_i^{(\ell)} \in \mathcal{Y}_\ell$, for all $\ell \in [L]$, the OMP algorithm selects points from $\mathcal{Y}_\ell$ in all $s_{\max}$ iterations.[3] Now, the OMP selection rule (2.1) for $\mathbf{y}_i^{(\ell)}$

[3] For DI-stopping the OMP algorithm terminates w.p. 1 after exactly $s_{\max}$ iterations as in our data model the points in $\mathcal{Y}$ are in general position w.p. 1 and $s_{\max} < \min_{\ell \in [L]} c_s d_\ell < \min\{m, N-1\}$ by the condition on $s_{\max}$ in Theorem 2.1.

implies that OMP selects a point from $\mathcal{Y}_\ell$ in the $(s+1)$-st iteration if

$$\max_{k\neq\ell,j}\left|\left\langle \mathbf{y}_j^{(k)}, \mathbf{r}_s\right\rangle\right| < \max_{j\in[n_\ell]\backslash(\Lambda_s\cup\{i\})}\left|\left\langle \mathbf{y}_j^{(\ell)}, \mathbf{r}_s\right\rangle\right|, \tag{2.16}$$

where $\max_{k\neq\ell,j}$ denotes maximization over subspaces $k\in[L]$, $k\neq\ell$, and over the indices j of the points $\mathbf{y}_j^{(k)}\in\mathcal{Y}_k$ in these subspaces. Hence, the graph G obtained by SSC-OMP satisfies the NFC property if (2.16) holds for all $s_{\max}$ iterations, for each $\mathbf{y}_i^{(\ell)}\in\mathcal{Y}_\ell$, for all $\ell\in[L]$. We now establish that this holds for our statistical data model w.p. at least $P^\star$ as defined in (2.9).

Our analysis will be based on an auxiliary algorithm termed "reduced OMP", which has access to the reduced dictionary $\mathcal{Y}_\ell\backslash\{\mathbf{y}_i^{(\ell)}\}$ only—instead of the full dictionary $\mathcal{Y}\backslash\{\mathbf{y}_i^{(\ell)}\}$—to represent $\mathbf{y}_i^{(\ell)}$. We henceforth use the shorthands $\mathbf{r}_s^{(\ell)}$ for the residuals $\mathbf{r}_s(\mathbf{y}_i^{(\ell)},\mathbf{Y}_{-i}^{(\ell)})$ corresponding to reduced OMP. The dependence of the $\mathbf{r}_s^{(\ell)}$ on the index i of the data point $\mathbf{y}_i^{(\ell)}$ to be represented is not made explicit for notational ease. If the $\mathbf{r}_s^{(\ell)}$ satisfy (2.16) for *all iterations* $s\in[s_{\max}]$, the reduced OMP algorithm and the original OMP algorithm select exactly the same data points and do so in exactly the same order. In this case, we also have $\mathbf{r}_s^{(\ell)}=\mathbf{r}_s$, for all $s\in[s_{\max}]$. As $\mathbf{r}_s^{(\ell)}$ satisfying (2.16) for all $s\in[s_{\max}]$ is necessary and sufficient for $\mathbf{r}_s$ to satisfy (2.16) for all $s\in[s_{\max}]$, a lower bound $P^\star$ on the probability of $\mathbf{r}_s^{(\ell)}$ satisfying (2.16) for all $s\in[s_{\max}]$ also constitutes a lower bound on the probability of $\mathbf{r}_s$ satisfying (2.16) for all $s\in[s_{\max}]$. Working with the reduced OMP algorithm is beneficial as $\mathbf{r}_s^{(\ell)}$ is a function of the points in $\mathcal{Y}_\ell$ only and is therefore statistically independent of the points in $\mathcal{Y}\backslash\mathcal{Y}_\ell$. This is significant as it will allow us to apply standard concentration of measure inequalities for independent random variables. In the remainder of the proof, we work with reduced OMP exclusively.

We start by providing intuition on the proof idea. To this end, we expand the inner products in (2.16) according to

$$\left\langle \mathbf{y}_j^{(k)}, \mathbf{r}_s^{(\ell)}\right\rangle = \left\langle \mathbf{x}_j^{(k)}+\mathbf{z}_j^{(k)}, \mathbf{r}_{s\|}^{(\ell)}+\mathbf{r}_{s\perp}^{(\ell)}\right\rangle$$

$$= \left\langle \mathbf{x}_j^{(k)}, \mathbf{r}_{s\|}^{(\ell)} \right\rangle + \left\langle \mathbf{x}_j^{(k)}, \mathbf{r}_{s\perp}^{(\ell)} \right\rangle \\ + \left\langle \mathbf{z}_j^{(k)}, \mathbf{r}_{s\|}^{(\ell)} \right\rangle + \left\langle \mathbf{z}_j^{(k)}, \mathbf{r}_{s\perp}^{(\ell)} \right\rangle . \tag{2.17}$$

The first term in (2.17) quantifies the similarity of the portions of $\mathbf{y}_j^{(k)}$ and $\mathbf{r}_s^{(\ell)}$ that lie in $\mathcal{S}_k$ and $\mathcal{S}_\ell$, respectively, i.e., the "signal components" of $\mathbf{y}_j^{(k)}$ and $\mathbf{r}_s^{(\ell)}$, while the other terms all account for interactions with or between "undesired components" residing in $\mathcal{S}_k^\perp$, $\mathcal{S}_\ell^\perp$. If the similarities—in absolute value—of the "signal components" for $k = \ell$, $j \in [n_\ell]\backslash(\Lambda_s \cup \{i\})$, are sufficiently large relative to those for $k \neq \ell$, $j \in [n_k]$, and if the interactions of all "undesired components" are sufficiently small, for $k, \ell \in [L]$, then (2.16) holds. Following (Heckel et al., 2017, proofs of Thm. 3, Cor. 1) this intuition will be made quantitative and rigorous by introducing events that, when conditioned on, yield bounds on the absolute values of the individual terms in (2.17) that are of analytically amenable form. These bounds will then be employed to derive an upper bound on the LHS and a lower bound on the RHS of (2.16) that both hold conditionally on the intersection of the underlying events. Based on these bounds, we will then show that the clustering condition (2.8) implies (2.16) w.p. at least $P^\star$. The particular choice of the events that we condition on is delicate, but when done properly, allows us to make the statistical dependencies between $\mathbf{r}_s^{(\ell)}$ and $\mathbf{y}_j^{(\ell)}$, $j \in [n_\ell]\backslash\{i\}$, analytically tractable. We finally note that although the general idea of conditioning on suitably defined events is taken from previous work by the authors (Heckel et al., 2017, proofs of Thm. 3, Cor. 1), the choice of the specific events as well as other technical aspects of the present proof differ significantly from (Heckel et al., 2017, proofs of Thm. 3, Cor. 1).

We commence the formal proof by upper-bounding the LHS of (2.16) according to

$$\max_{k\neq\ell,j} \left| \left\langle \mathbf{x}_j^{(k)} + \mathbf{z}_j^{(k)}, \mathbf{r}_{s\|}^{(\ell)} + \mathbf{r}_{s\perp}^{(\ell)} \right\rangle \right| \\ \leq \max_{k\neq\ell,j} \left| \left\langle \mathbf{x}_j^{(k)}, \mathbf{r}_{s\|}^{(\ell)} \right\rangle \right| + \max_{k\neq\ell,j} \left| \left\langle \mathbf{x}_j^{(k)}, \mathbf{r}_{s\perp}^{(\ell)} \right\rangle \right|$$

$$
\begin{aligned}
&+\max_{k\neq\ell,j}\left|\left\langle \mathbf{z}_j^{(k)},\mathbf{r}_s^{(\ell)}\right\rangle\right| \\
&\le 4\log(N^3 s_{\max})\frac{\left\|{\mathbf{U}^{(k)}}^\top\mathbf{U}^{(\ell)}\right\|_F}{\sqrt{d_k}\sqrt{d_\ell}}\left\|\mathbf{r}_{s\parallel}^{(\ell)}\right\|_2 \\
&\quad+\frac{\sqrt{2\log(N^3 s_{\max})}}{\sqrt{m-d_\ell}}\left\|\mathbf{r}_{s\perp}^{(\ell)}\right\|_2 \\
&\quad+\frac{\sqrt{2\log(N^3 s_{\max})}}{\sqrt{m}}\frac{3}{2}\sigma\left(1+\frac{3}{2}\sigma\right),
\end{aligned}
\tag{2.18}
$$

where the second inequality holds on the event $\mathcal{E}_1^{(\ell,i,s)}\cap\mathcal{E}_2^{(\ell,i,s)}\cap\mathcal{E}_3^{(\ell,i,s)}\cap\mathcal{E}_4$ with

$$
\begin{aligned}
\mathcal{E}_1^{(\ell,i,s)} :=& \left\{\max_{k\neq\ell,j}\left|\left\langle \mathbf{x}_j^{(k)},\mathbf{r}_{s\parallel}^{(\ell)}\right\rangle\right|\right. \\
&\left.\le 4\log(N^3 s_{\max})\frac{\left\|{\mathbf{U}^{(k)}}^\top\mathbf{U}^{(\ell)}\right\|_F}{\sqrt{d_k}\sqrt{d_\ell}}\left\|\mathbf{r}_{s\parallel}^{(\ell)}\right\|_2\right\},
\end{aligned}
\tag{2.19}
$$

$$
\mathcal{E}_2^{(\ell,i,s)} := \left\{\max_{k\neq\ell,j}\left|\left\langle \mathbf{x}_j^{(k)},\mathbf{r}_{s\perp}^{(\ell)}\right\rangle\right|\le\frac{\sqrt{2\log(N^3 s_{\max})}}{\sqrt{m-d_\ell}}\left\|\mathbf{r}_{s\perp}^{(\ell)}\right\|_2\right\},
\tag{2.20}
$$

$$
\begin{aligned}
\mathcal{E}_3^{(\ell,i,s)} :=& \left\{\max_{k\neq\ell,j}\left|\left\langle \mathbf{z}_j^{(k)},\mathbf{r}_s^{(\ell)}\right\rangle\right|\right. \\
&\left.\le\frac{\sqrt{2\log(N^3 s_{\max})}}{\sqrt{m}}\left(1+\left\|\mathbf{z}_i^{(\ell)}\right\|_2\right)\max_{k\neq\ell,j}\left\|\mathbf{z}_j^{(k)}\right\|_2\right\},
\end{aligned}
\tag{2.21}
$$

$$
\begin{aligned}
\mathcal{E}_4 :=& \left\{\left\{\left\|\mathbf{z}_{j\parallel}^{(\ell)}\right\|_2\le\frac{3}{2}\frac{\sqrt{d_\ell}}{\sqrt{m}}\sigma\right\}\cap\left\{\left\|\mathbf{z}_j^{(\ell)}\right\|_2\le\frac{3}{2}\sigma\right\},\right. \\
&\left.\forall\ell\in[L],j\in[n_\ell]\right\}.
\end{aligned}
\tag{2.22}
$$

Note that the dependence of $\mathcal{E}_1^{(\ell,i,s)}$ and $\mathcal{E}_2^{(\ell,i,s)}$ on i is due to $\mathbf{r}_{s\parallel}^{(\ell)}$ and $\mathbf{r}_{s\perp}^{(\ell)}$, both of which are functions of $\mathbf{y}_i^{(\ell)}$. Here, $\mathcal{E}_1^{(\ell,i,s)}$ pertains to the similarities of the "signal components" for $k \neq \ell$, $\mathcal{E}_2^{(\ell,i,s)}$ and $\mathcal{E}_3^{(\ell,i,s)}$ quantify the similarity of "undesired components", and $\mathcal{E}_4$ controls the magnitude of the "undesired components" of the $\mathbf{y}_j^{(\ell)}$.

We proceed by lower-bounding the RHS of (2.16). Using $\|\mathbf{r}_s^{(\ell)}\|_2 = \|(\mathbf{I} - \mathbf{Y}_{\Lambda_s}^{(\ell)}(\mathbf{Y}_{\Lambda_s}^{(\ell)})^\dagger)\mathbf{y}_i^{(\ell)}\|_2 \leq \|\mathbf{y}_i^{(\ell)}\|_2 \leq 1 + \|\mathbf{z}_i^{(\ell)}\|_2$ (where the inequality is thanks to $\|\mathbf{x}_i^{(\ell)}\|_2 = 1$ and $\mathbf{I} - \mathbf{Y}_{\Lambda_s}^{(\ell)}(\mathbf{Y}_{\Lambda_s}^{(\ell)})^\dagger$ being an orthogonal projection matrix), we find that on the event $\mathcal{E}_4 \cap \mathcal{E}_5^{(\ell,i)}$ with

$$\begin{aligned}\mathcal{E}_5^{(\ell,i)} := \Bigg\{ \max_{j \in [n_\ell] \setminus (\Lambda_s \cup \{i\})} \left| \left\langle \mathbf{y}_j^{(\ell)}, \mathbf{r}_s^{(\ell)} \right\rangle \right| &\\ \geq \left(1 - \frac{c_4 + 1}{\sqrt{\rho_\ell}}\right) \frac{\left\|\mathbf{r}_{s\|}^{(\ell)}\right\|_2}{\sqrt{d_\ell}} &\\ - \sigma \left(\frac{1}{\sqrt{m}} + \frac{2}{\sqrt{n_\ell - 1}}\right) \left\|\mathbf{r}_s^{(\ell)}\right\|_2 \Bigg\}, &\end{aligned} \tag{2.23}$$

where $c_4 > 0$ is the numerical constant in Lemma 2.9, the RHS of (2.16) obeys

$$\begin{aligned}\max_{j \in [n_\ell] \setminus (\Lambda_s \cup \{i\})} \left| \left\langle \mathbf{y}_j^{(\ell)}, \mathbf{r}_s^{(\ell)} \right\rangle \right| \geq \left(1 - \frac{c_4 + 1}{\sqrt{\rho_\ell}}\right) \frac{\left\|\mathbf{r}_{s\|}^{(\ell)}\right\|_2}{\sqrt{d_\ell}} &\\ - \sigma \left(\frac{1}{\sqrt{m}} + \frac{2}{\sqrt{n_\ell - 1}}\right) \left(1 + \frac{3}{2}\sigma\right). &\end{aligned} \tag{2.24}$$

On $\mathcal{E}_1^{(\ell,i,s)} \cap \mathcal{E}_2^{(\ell,i,s)} \cap \mathcal{E}_3^{(\ell,i,s)} \cap \mathcal{E}_4 \cap \mathcal{E}_5^{(\ell,i)}$, (2.16) is now implied by [RHS of (2.18)] < [RHS of (2.24)]; multiplying this inequality by $\sqrt{d_\ell}/(4\log(N^3 s_{\max})\|\mathbf{r}_{s\|}^{(\ell)}\|_2)$, we get

$$\underbrace{\frac{\left\|\mathbf{U}^{(k)\top}\mathbf{U}^{(\ell)}\right\|_F}{\sqrt{d_k}}}_{\leq \max\limits_{k:\, k \neq \ell} \operatorname{aff}(\mathcal{S}_k, \mathcal{S}_\ell)} + \frac{1}{\sqrt{8\log(N^3 s_{\max})}} \left(\underbrace{\frac{\sqrt{d_\ell}}{\sqrt{m - d_\ell}}}_{\leq \sqrt{2d_\ell}/\sqrt{m}} \frac{\left\|\mathbf{r}_{s\perp}^{(\ell)}\right\|_2}{\left\|\mathbf{r}_{s\|}^{(\ell)}\right\|_2}\right.$$

$$
\qquad + \frac{\sigma}{\left\|\mathbf{r}_{s\|}^{(\ell)}\right\|_2} \frac{\sqrt{d_\ell}}{\sqrt{m}} \frac{3}{2}\left(1+\frac{3}{2}\sigma\right)\Bigg)
$$

$$
< \frac{1}{4\log(N^3 s_{\max})}\Bigg(\underbrace{\left(1-\frac{c_4+1}{\sqrt{\rho_\ell}}\right)}_{\geq 1/2}
$$

$$
- \frac{\sigma}{\left\|\mathbf{r}_{s\|}^{(\ell)}\right\|_2}\left(\frac{\sqrt{d_\ell}}{\sqrt{m}} + \underbrace{\frac{2\sqrt{d_\ell}}{\sqrt{n_\ell - 1}}}_{=2/\sqrt{\rho_\ell}}\right)\left(1+\frac{3}{2}\sigma\right)\Bigg), \tag{2.25}
$$

where $1-(c_4+1)/\sqrt{\rho_\ell} \geq 1/2$ follows from $\rho_\ell \geq \rho_{\min} \geq c_\rho := 4(c_4+1)^2$, for all $\ell \in [L]$, and $\sqrt{d_\ell}/\sqrt{m-d_\ell} \leq \sqrt{2d_\ell}/\sqrt{m}$ is by $m \geq 2d_{\max} \geq 2d_\ell$, for all $\ell \in [L]$. Rearranging terms in (2.25) and using $\sqrt{8\log(N^3 s_{\max})} < 4\log(N^3 s_{\max})$, for $N \geq 2$, we can see that (2.25) is implied by

$$
\max_{k\colon k\neq\ell} \operatorname{aff}(\mathcal{S}_k,\mathcal{S}_\ell) + \frac{1}{\left\|\mathbf{r}_{s\|}^{(\ell)}\right\|_2 \sqrt{8\log(N^3 s_{\max})}}\Bigg(\frac{\sqrt{2d_\ell}}{\sqrt{m}}\left\|\mathbf{r}_{s\perp}^{(\ell)}\right\|_2
$$

$$
+\sigma\left(\frac{5}{2}\frac{\sqrt{d_\ell}}{\sqrt{m}} + \frac{2}{\sqrt{\rho_\ell}}\right)\left(1+\frac{3}{2}\sigma\right)\Bigg) \leq \frac{1}{8\log(N^3 s_{\max})}. \tag{2.26}
$$

To further simplify (2.26), we upper-bound $\|\mathbf{r}_{s\perp}^{(\ell)}\|_2$ and lower-bound $\|\mathbf{r}_{s\|}^{(\ell)}\|_2$. To this end, we introduce the events

$$
\begin{aligned}
\mathcal{E}_6^{(\ell,i)} &:= \left\{\left\|\mathbf{r}_{s\perp}^{(\ell)}\right\|_2 \leq \left\|\mathbf{z}_{i\perp}^{(\ell)}\right\|_2 + \frac{3\sigma}{\tilde{a}}\left\|\mathbf{y}_i^{(\ell)}\right\|_2, \forall s \leq s_{\max}\right\}, \\
\mathcal{E}_7^{(\ell,i)} &:= \left\{\left\|\mathbf{r}_{s\|}^{(\ell)}\right\|_2 > \left\|\mathbf{y}_{i\|}^{(\ell)}\right\|_2\left(\frac{2}{3} - \sqrt{\frac{3\bar{s}\log((n_\ell-1)e/\bar{s})}{d_\ell}}\right), \forall s \leq \bar{s}\right\},
\end{aligned} \tag{2.27}
$$

where $\tilde{a} := \min_{j\in[n_\ell]\setminus\{i\}} \|\mathbf{y}_{j\|}^{(\ell)}\|_2 \geq \min_{j\in[n_\ell]\setminus\{i\}}(\|\mathbf{x}_j^{(\ell)}\|_2 - \|\mathbf{z}_{j\|}^{(\ell)}\|_2) \geq 1 - \max_{j\in[n_\ell]\setminus\{i\}} \|\mathbf{z}_{j\|}^{(\ell)}\|_2$. Setting $\bar{s} = s_{\max}$ in (2.27), on $\mathcal{E}_4 \cap \mathcal{E}_6^{(\ell,i)} \cap$

$\mathcal{E}_7^{(\ell,i)}$, we have

$$\left\|\mathbf{r}_{s\perp}^{(\ell)}\right\|_2 \leq \frac{3}{2}\sigma + 3\sigma\frac{1+\frac{3}{2}\sigma}{1-\frac{3}{2}\frac{\sqrt{d_\ell}}{\sqrt{m}}\sigma} \leq \sigma(8+10\sigma), \tag{2.28}$$

$$\begin{aligned}\left\|\mathbf{r}_{s\|}^{(\ell)}\right\|_2 &> \left(1-\frac{3}{2}\frac{\sqrt{d_\ell}}{\sqrt{m}}\sigma\right)\left(\frac{2}{3}-\sqrt{3c_s}\right)\\ &> \left(1-\frac{3}{2}\frac{\sqrt{d_\ell}}{\sqrt{m}}\sigma\right)\frac{1}{9} > \frac{1}{20},\end{aligned} \tag{2.29}$$

where we employed the assumptions $m \geq 2d_{\max} \geq 2d_\ell$, for all $\ell \in [L]$, and $\sigma \leq 1/2$ to get (2.28), and used $s_{\max} \leq c_s d_\ell / \log((n_\ell - 1)e/s_{\max})$, for all $\ell \in [L]$, and $c_s := \min\{1/10, c_1\}$ (with c_1 the constant in Lemma 2.7 below), to arrive at (2.29). With (2.28) and (2.29), it follows that (2.26) is implied by

$$\max_{k:\, k\neq\ell} \operatorname{aff}(\mathcal{S}_k,\mathcal{S}_\ell) + \frac{10\sigma}{\sqrt{\log(N^3 s_{\max})}}\left(\frac{\sqrt{d_\ell}}{\sqrt{m}}(10+13\sigma) + \frac{\sqrt{2}}{\sqrt{\rho_\ell}}\left(1+\frac{3}{2}\sigma\right)\right) \leq \frac{1}{8\log(N^3 s_{\max})}.$$

This inequality holds for all $\ell \in [L]$ by the clustering condition (2.8) with $c(\sigma) = 10 + 13\sigma$. Hence, on the event

$$\mathcal{E}^\star := \bigcap_{\ell,i,s}\big(\mathcal{E}_1^{(\ell,i,s)} \cap \mathcal{E}_2^{(\ell,i,s)} \cap \mathcal{E}_3^{(\ell,i,s)} \cap \mathcal{E}_4 \cap \mathcal{E}_5^{(\ell,i)} \cap \mathcal{E}_6^{(\ell,i)} \cap \mathcal{E}_7^{(\ell,i)}\big), \tag{2.30}$$

(2.8) implies (2.16) for every $\mathbf{y}_i^{(\ell)} \in \mathcal{Y}_\ell$, for all $\ell \in [L]$, and the graph G obtained by SSC-OMP has no false connections. It remains to lower-bound $\mathrm{P}[\mathcal{E}^\star]$. By the union bound, we have

$$\begin{aligned}\mathrm{P}[\mathcal{E}^\star] &= 1 - \mathrm{P}\left[\overline{\mathcal{E}^\star}\right]\\ &\geq 1 - \mathrm{P}\left[\overline{\mathcal{E}}_4\right]\end{aligned}$$

$$
\begin{aligned}
&- \sum_{\ell\in[L],i\in[n_\ell]} \left(\mathrm{P}\left[\overline{\mathcal{E}}_5^{(\ell,i)}\right] + \mathrm{P}\left[\overline{\mathcal{E}}_6^{(\ell,i)}\right] + \mathrm{P}\left[\overline{\mathcal{E}}_7^{(\ell,i)}\right]\right) \\
&- \sum_{\substack{\ell\in[L],i\in[n_\ell],\\ s\in[s_{\max}]}} \left(\mathrm{P}\left[\overline{\mathcal{E}}_1^{(\ell,i,s)}\right] + \mathrm{P}\left[\overline{\mathcal{E}}_2^{(\ell,i,s)}\right] + \mathrm{P}\left[\overline{\mathcal{E}}_3^{(\ell,i,s)}\right]\right) \\
\geq{}& 1 - \sum_{\ell\in[L]} n_\ell(e^{-d_\ell/8} + e^{-m/8}) \\
&- \sum_{\ell\in[L],i\in[n_\ell]} \Big(2e^{-c_5 d_\ell} + 2e^{-2m} \\
&\qquad + 2e^{-c_2 m} + 2e^{-c_3 d_\ell} + e^{-d_\ell/18}\Big) \\
&- \sum_{\substack{\ell\in[L],i\in[n_\ell],\\ s\in[s_{\max}]}} \left(\frac{2}{N^2 s_{\max}} + \frac{2}{N^2 s_{\max}} + \frac{2}{N^2 s_{\max}}\right) \qquad (2.31) \\
\geq{}& 1 - \frac{6}{N} - \sum_{\ell\in[L]} n_\ell(6e^{-c_d d_\ell} + 5e^{-c_m m}),
\end{aligned}
$$

where $c_d := \min\{1/18, c_3, c_5\}$, $c_m := \min\{1/8, c_2\}$, and (2.31) follows from Lemmata 2.2, 2.5, 2.7, 2.9, and 2.10.

The proofs of Lemmata 2.2 and 2.5 rely on the rotational invariance of the distributions of $\mathbf{r}_{s\|}^{(\ell)}$ and $\mathbf{r}_{s\perp}^{(\ell)}$ on $\mathcal{S}_\ell$ and $\mathcal{S}_\ell^\perp$, respectively, which is inherited from the rotational invariance of the distributions of the $\mathbf{x}_j^{(\ell)}$ and $\mathbf{z}_j^{(\ell)}$ characterized next.

Lemma 2.1. *The distributions of* $\mathbf{r}_{s\|}^{(\ell)}$ *and* $\mathbf{r}_{s\perp}^{(\ell)}$ *are rotationally invariant on* $\mathcal{S}_\ell$ *and* $\mathcal{S}_\ell^\perp$*, respectively, for all* $i \in [n_\ell]$, $\ell \in [L]$, *i.e., we have* $\mathbf{V}^\| \mathbf{r}_{s\|}^{(\ell)} \sim \mathbf{r}_{s\|}^{(\ell)}$ *and* $\mathbf{V}^\perp \mathbf{r}_{s\perp}^{(\ell)} \sim \mathbf{r}_{s\perp}^{(\ell)}$ *for all unitary matrices* $\mathbf{V}^\|$ *and* $\mathbf{V}^\perp$ *of the form* $\mathbf{V}^\| = \mathbf{U}^{(\ell)}\mathbf{W}^\|{\mathbf{U}^{(\ell)}}^\top + \mathbf{P}_\perp$ *and* $\mathbf{V}^\perp = \mathbf{P}_\| + \mathbf{U}_o^{(\ell)}\mathbf{W}^\perp{\mathbf{U}_o^{(\ell)}}^\top$*, respectively, where* $\mathbf{W}^\| \in \mathbb{R}^{d_\ell\times d_\ell}$, $\mathbf{W}^\perp \in \mathbb{R}^{(m-d_\ell)\times(m-d_\ell)}$ *are unitary and the columns of* $\mathbf{U}_o^{(\ell)} \in \mathbb{R}^{m\times(m-d_\ell)}$ *form an orthonormal basis for* $\mathcal{S}_\ell^\perp$*.*

Note that the unitary transformations $\mathbf{V}^\|$ and $\mathbf{V}^\perp$ act only on $\mathcal{S}_\ell$ and $\mathcal{S}_\ell^\perp$, respectively, and leave components in $\mathcal{S}_\ell^\perp$ and $\mathcal{S}_\ell$, respectively, unchanged.

Proof. We first show that the reduced OMP residual $\mathbf{r}_s^{(\ell)}$ is covariant w.r.t. transformations of the points in $\mathcal{Y}_\ell$ by a unitary matrix $\mathbf{V} \in \mathbb{R}^{m\times m}$, i.e., we establish that $\mathbf{r}_s(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)}) = \mathbf{V}\mathbf{r}_s(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)})$, for all $s \in [s_{\max}]$, $i \in [n_\ell]$, $\ell \in [L]$. Combining this covariance property of $\mathbf{r}_s^{(\ell)}$ with the rotational invariance on $\mathcal{S}_\ell$ and $\mathcal{S}_\ell^\perp$ of the distributions of $\mathbf{y}_{j\|}^{(\ell)} = \mathbf{U}^{(\ell)}\tilde{\mathbf{a}}_j^{(\ell)}$ and $\mathbf{y}_{j\perp}^{(\ell)} = \mathbf{P}_\perp\mathbf{z}_j^{(\ell)}$, respectively, then establishes the desired result.

We prove $\mathbf{r}_s(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)}) = \mathbf{V}\mathbf{r}_s(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)})$ by induction and start with the inductive step. Assume that after some iteration $s' < s_{\max}$, the index set $\Lambda_{s'}$ corresponding to the transformed data $(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)})$ is identical to the index set $\Lambda_{s'}$ associated with the original data, i.e., $\Lambda_{s'}(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)}) = \Lambda_{s'}(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)})$. Using the shorthands $\Lambda_{s'}(\mathbf{V})$ for $\Lambda_{s'}(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)})$ and $\Lambda_{s'}$ for $\Lambda_{s'}(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)})$, it then follows that

$$\begin{aligned}
\mathbf{r}_{s'}(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)}) &= \left(\mathbf{I} - \mathbf{V}\mathbf{Y}_{\Lambda_{s'}(\mathbf{V})}^{(\ell)}\left(\mathbf{V}\mathbf{Y}_{\Lambda_{s'}(\mathbf{V})}^{(\ell)}\right)^{\dagger}\right)\mathbf{V}\mathbf{y}_i^{(\ell)} \\
&= \left(\mathbf{I} - \mathbf{V}\mathbf{Y}_{\Lambda_{s'}}^{(\ell)}\left(\mathbf{V}\mathbf{Y}_{\Lambda_{s'}}^{(\ell)}\right)^{\dagger}\right)\mathbf{V}\mathbf{y}_i^{(\ell)} \\
&= \left(\mathbf{I} - \mathbf{V}\mathbf{Y}_{\Lambda_{s'}}^{(\ell)}\left({\mathbf{Y}_{\Lambda_{s'}}^{(\ell)}}^{\top}\mathbf{V}^{\top}\mathbf{V}\mathbf{Y}_{\Lambda_{s'}}^{(\ell)}\right)^{-1}{\mathbf{Y}_{\Lambda_{s'}}^{(\ell)}}^{\top}\mathbf{V}^{\top}\right)\mathbf{V}\mathbf{y}_i^{(\ell)} \\
&= \mathbf{V}\left(\mathbf{I} - \mathbf{Y}_{\Lambda_{s'}}^{(\ell)}\left({\mathbf{Y}_{\Lambda_{s'}}^{(\ell)}}^{\top}\mathbf{Y}_{\Lambda_{s'}}^{(\ell)}\right)^{-1}{\mathbf{Y}_{\Lambda_{s'}}^{(\ell)}}^{\top}\right)\mathbf{y}_i^{(\ell)} \\
&= \mathbf{V}\mathbf{r}_{s'}(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)}). \qquad (2.32)
\end{aligned}$$

For the index $\lambda_{s'+1}(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)})$ selected for the $\mathbf{V}$-transformed data set in iteration $s'+1$, (2.32) implies

$$\begin{aligned}
\lambda_{s'+1}(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)}) &= \underset{j\in[n_\ell]\setminus(\Lambda_{s'}\cup\{i\})}{\arg\max}\left|\left\langle \mathbf{V}\mathbf{y}_j^{(\ell)}, \mathbf{r}_{s'}(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)})\right\rangle\right| \\
&= \underset{j\in[n_\ell]\setminus(\Lambda_{s'}\cup\{i\})}{\arg\max}\left|\left\langle \mathbf{y}_j^{(\ell)}, \mathbf{r}_{s'}(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)})\right\rangle\right|
\end{aligned}$$

$$= \lambda_{s'+1}(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)}), \tag{2.33}$$

i.e., the index selected in iteration $s'+1$ by operating on the $\mathbf{V}$-transformed data set is identical to that obtained for the original data set. It remains to establish the base case. This is done by noting that thanks to $\mathbf{r}_0(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)}) = \mathbf{V}\mathbf{y}_i^{(\ell)}$, we have

$$\begin{aligned}
\lambda_1(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)}) &= \underset{j\in[n_\ell]\setminus(\Lambda_{s'}\cup\{i\})}{\arg\max} \left|\langle \mathbf{V}\mathbf{y}_j^{(\ell)}, \mathbf{V}\mathbf{y}_i^{(\ell)}\rangle\right| \\
&= \underset{j\in[n_\ell]\setminus(\Lambda_{s'}\cup\{i\})}{\arg\max} \left|\langle \mathbf{y}_j^{(\ell)}, \mathbf{y}_i^{(\ell)}\rangle\right| \\
&= \lambda_1(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)}).
\end{aligned}$$

We next establish the rotational invariance of $\mathbf{r}_{s\|}^{(\ell)}$ and $\mathbf{r}_{s\perp}^{(\ell)}$. Note that $\mathbf{V}^{\|}\mathbf{y}_j^{(\ell)} = \mathbf{U}^{(\ell)}\mathbf{W}^{\|}{\mathbf{U}^{(\ell)}}^{\top}\mathbf{y}_{j\|}^{(\ell)} + \mathbf{y}_{j\perp}^{(\ell)} = \mathbf{U}^{(\ell)}\mathbf{W}^{\|}\tilde{\mathbf{a}}_j^{(\ell)} + \mathbf{y}_{j\perp}^{(\ell)} \sim \mathbf{U}^{(\ell)}\tilde{\mathbf{a}}_j^{(\ell)} + \mathbf{y}_{j\perp}^{(\ell)} = \mathbf{y}_j^{(\ell)}, j \in [n_\ell]$, and $\mathbf{V}^{\perp}\mathbf{y}_j^{(\ell)} = \mathbf{y}_{j\|}^{(\ell)} + \mathbf{U}_o^{(\ell)}\mathbf{W}^{\perp}{\mathbf{U}_o^{(\ell)}}^{\top}\mathbf{y}_{j\perp}^{(\ell)} = \mathbf{y}_{j\|}^{(\ell)} + \mathbf{U}_o^{(\ell)}\mathbf{W}^{\perp}{\mathbf{U}_o^{(\ell)}}^{\top}\mathbf{z}_{j\perp}^{(\ell)} \sim \mathbf{y}_{j\|}^{(\ell)} + \mathbf{z}_{j\perp}^{(\ell)} = \mathbf{y}_j^{(\ell)}, j \in [n_\ell]$, by unitarity of $\mathbf{V}^{\|}$ and $\mathbf{V}^{\perp}$. Together with $\mathbf{r}_s(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)}) = \mathbf{V}\mathbf{r}_s(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)})$, this yields

$$\begin{aligned}
\mathbf{r}_{s\|}(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)}) &= \mathbf{P}_{\|}\mathbf{r}_s(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)}) && (2.34)\\
&\sim \mathbf{P}_{\|}\mathbf{r}_s(\mathbf{V}^{\|}\mathbf{y}_i^{(\ell)}, \mathbf{V}^{\|}\mathbf{Y}_{-i}^{(\ell)}) \\
&= \mathbf{P}_{\|}\mathbf{V}^{\|}\mathbf{r}_s(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)}) \\
&= (\underbrace{\mathbf{P}_{\|}\mathbf{U}^{(\ell)}\mathbf{W}^{\|}{\mathbf{U}^{(\ell)}}^{\top}}_{=\mathbf{U}^{(\ell)}\mathbf{W}^{\|}{\mathbf{U}^{(\ell)}}^{\top}\mathbf{P}_{\|}} + \underbrace{\mathbf{P}_{\|}\mathbf{P}_{\perp}}_{=\mathbf{0}})\mathbf{r}_s(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)}) \\
&= \mathbf{V}^{\|}\mathbf{P}_{\|}\mathbf{r}_s(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)}) \\
&= \mathbf{V}^{\|}\mathbf{r}_{s\|}(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)}), && (2.35)
\end{aligned}$$

and establishes $\mathbf{V}^{\|}\mathbf{r}_{s\|} \sim \mathbf{r}_{s\|}$. Repeating the steps leading from (2.34) to (2.35) with $\mathbf{P}_{\perp}$ and $\mathbf{V}^{\perp}$ in place of $\mathbf{P}_{\|}$ and $\mathbf{V}^{\|}$, respectively, we analogously obtain $\mathbf{r}_{s\perp} \sim \mathbf{V}^{\perp}\mathbf{r}_{s\perp}$, thereby finishing the proof. □

We next derive lower bounds on $\mathrm{P}[\mathcal{E}_1^{(\ell,i,s)}]$, $\mathrm{P}[\mathcal{E}_2^{(\ell,i,s)}]$, and $\mathrm{P}[\mathcal{E}_3^{(\ell,i,s)}]$.

Lemma 2.2. *We have*

$$\mathrm{P}\left[\mathcal{E}_1^{(\ell,i,s)}\right] \geq 1 - \frac{2}{N^2 s_{\max}}, \quad \mathrm{P}\left[\mathcal{E}_2^{(\ell,i,s)}\right] \geq 1 - \frac{2}{N^2 s_{\max}},$$
$$\mathrm{P}\left[\mathcal{E}_3^{(\ell,i,s)}\right] \geq 1 - \frac{2}{N^2 s_{\max}}. \tag{2.36}$$

Proof. First note that $\mathbf{r}_{s\|}^{(\ell)}/\|\mathbf{r}_{s\|}^{(\ell)}\|_2$ and $\mathbf{r}_{s\perp}^{(\ell)}/\|\mathbf{r}_{s\perp}^{(\ell)}\|_2$ are distributed uniformly at random on $\mathbb{S}^{m-1} \cap \mathcal{S}_\ell$ and $\mathbb{S}^{m-1} \cap \mathcal{S}_\ell^\perp$, respectively, as a consequence of rotational invariance (Lemma 2.1) and normalization (Muirhead, 2009, Thm. 1.5.6). This allows us to apply Lemma 2.3 below with $\mathbf{L} = \mathbf{A}^{(k)}$, $\mathbf{C} = {\mathbf{U}^{(k)}}^\top \mathbf{U}^{(\ell)}$, $\mathbf{a} = \mathbf{r}_{s\|}^{(\ell)}/\|\mathbf{r}_{s\|}^{(\ell)}\|_2$, and $\alpha = 4\log(N^3 s_{\max})$ (note that the condition $\alpha > 12$ is satisfied as $N \geq 3$ and $s_{\max} \geq 1$ by the assumptions of Theorem 2.1) to get a lower bound on $\mathrm{P}\left[\mathcal{E}_1^{(\ell,i,s)}\right]$ according to

$$\begin{aligned}\mathrm{P}\Bigg[\max_{j\in[n_k]} \left|\left\langle \mathbf{x}_j^{(k)}, \mathbf{r}_{s\|}^{(\ell)}\right\rangle\right| \\ > 4\log(N^3 s_{\max}) \frac{\left\|{\mathbf{U}^{(k)}}^\top \mathbf{U}^{(\ell)}\right\|_F}{\sqrt{d_k}\sqrt{d_\ell}} \left\|\mathbf{r}_{s\|}^{(\ell)}\right\|_2\Bigg] &\leq \frac{n_k+1}{N^3 s_{\max}} \\ &\leq \frac{2n_k}{N^3 s_{\max}},\end{aligned}$$

for $k \neq \ell$. The desired bound on $\mathrm{P}[\mathcal{E}_1^{(\ell,i,s)}]$ then follows by a union bound over $k \in [L]\backslash\{\ell\}$.

The lower bound on $\mathrm{P}[\mathcal{E}_2^{(\ell,i,s)}]$ is obtained by invoking Lemma 2.4 below with $\mathbf{a} = \mathbf{r}_{s\perp}^{(\ell)}/\|\mathbf{r}_{s\perp}^{(\ell)}\|_2 \in \mathcal{S}_\ell^\perp$ (hence replacing $\mathbb{S}^{m-1}$ by $\mathcal{S}_\ell^\perp \cap \mathbb{S}^{m-1}$), $\mathbf{b} = \mathbf{x}_j^{(k)}$, and $\beta = \sqrt{2\log(N^3 s_{\max})}$, which yields

$$\mathrm{P}\left[\left|\left\langle \mathbf{r}_{s\perp}^{(\ell)}, \mathbf{x}_j^{(k)}\right\rangle\right| > \frac{\sqrt{2\log(N^3 s_{\max})}}{\sqrt{m-d_\ell}} \left\|\mathbf{r}_{s\perp}^{(\ell)}\right\|_2\right] \leq \frac{2}{N^3 s_{\max}}, \tag{2.37}$$

for $k \neq \ell$. Again, a union bound over $k \in [L]\backslash\{\ell\}$, $j \in [n_k]$, gives the desired bound on $\mathrm{P}[\mathcal{E}_2^{(\ell,i,s)}]$.

Finally, for $\mathrm{P}[\mathcal{E}_3^{(\ell,i,s)}]$, we set $\mathbf{a} = \mathbf{z}_j^{(k)}/\|\mathbf{z}_j^{(k)}\|_2$, $\mathbf{b} = \mathbf{r}_s^{(\ell)}$, and $\beta = \sqrt{2\log(N^3 s_{\max})}$ in Lemma 2.4, and use $\|\mathbf{r}_s^{(\ell)}\|_2 \leq 1 + \|\mathbf{z}_i^{(\ell)}\|_2$, to obtain

$$\mathrm{P}\left[\left|\left\langle \mathbf{z}_j^{(k)}, \mathbf{r}_s^{(\ell)} \right\rangle\right| > \frac{\sqrt{2\log(N^3 s_{\max})}}{\sqrt{m}} \left(1 + \left\|\mathbf{z}_i^{(\ell)}\right\|_2\right) \left\|\mathbf{z}_j^{(k)}\right\|_2\right] \\ \leq \frac{2}{N^3 s_{\max}},$$

for all $k \neq \ell$. Again, the lower bound on $\mathrm{P}[\mathcal{E}_3^{(\ell,i,s)}]$ follows from a union bound over $k \in [L]\backslash\{\ell\}$, $j \in [n_k]$.

Lemma 2.3 (Extracted from the proof of (Soltanolkotabi and Candès, 2012, Lem. 7.5))**.** *Let* $\mathbf{a} \in \mathbb{R}^{d_2}$ *be distributed uniformly at random on* $\mathbb{S}^{d_2-1}$ *and let the columns of* $\mathbf{L} \in \mathbb{R}^{d_1 \times n_1}$ *be independent and distributed uniformly on* $\mathbb{S}^{d_1-1}$*. Let* $\mathbf{C} \in \mathbb{R}^{d_1 \times d_2}$*. Then, for* $\alpha \geq 12$*, we have*

$$\mathrm{P}\left[\|\mathbf{LCa}\|_\infty \geq \frac{\alpha}{\sqrt{d_1}\sqrt{d_2}}\|\mathbf{C}\|_F\right] \leq (n_1+1)e^{-\alpha/4}. \tag{2.38}$$

Lemma 2.4 (E.g., (Vershynin, 2012, Ex. 5.25))**.** *Let* $\mathbf{a}$ *be uniformly distributed on* $\mathbb{S}^{m-1}$ *and fix* $\mathbf{b} \in \mathbb{R}^m$*. Then, for* $\beta \geq 0$*, we have*

$$\mathrm{P}\left[|\langle \mathbf{a}, \mathbf{b}\rangle| > \frac{\beta}{\sqrt{m}}\|\mathbf{b}\|_2\right] \leq 2e^{-\frac{\beta^2}{2}}.$$

□

Next, we lower-bound $\mathrm{P}[\mathcal{E}_7^{(\ell,i)}]$.

Lemma 2.5. *Let* $n_\ell \geq d_\ell + 1$ *and* $\bar{s} \leq d_\ell$*. We have*

$$\mathrm{P}\left[\mathcal{E}_7^{(\ell,i)}\right] = \mathrm{P}\left[\left\|\mathbf{r}_{s\|}^{(\ell)}\right\|_2\right.$$

$$> \left\|\mathbf{y}_{i\|}^{(\ell)}\right\|_2 \left(\frac{2}{3} - \sqrt{\frac{3\bar{s}\log((n_\ell - 1)e/\bar{s})}{d_\ell}}\right), \ \forall s \leq \bar{s}\Bigg]$$
$$\geq 1 - e^{-d_\ell/18}. \qquad (2.39)$$

Proof. The bound obviously holds for $s = 0$ as $\|\mathbf{r}_{0\|}^{(\ell)}\| = \|\mathbf{y}_{i\|}^{(\ell)}\|_2$. For $1 \leq s \leq \bar{s}$ the outline of the proof is as follows. As $\|\mathbf{r}_{s\|}^{(\ell)}\|_2 = \|\mathbf{P}_{\|}(\mathbf{I} - \mathbf{Y}_{\Lambda_s}^{(\ell)}\mathbf{Y}_{\Lambda_s}^{(\ell)^\dagger})\mathbf{y}_i^{(\ell)}\|_2$ is hard to analyze directly owing to statistical dependencies between $\mathbf{y}_i^{(\ell)}$ and the columns of $\mathbf{Y}_{\Lambda_s}^{(\ell)}$ induced by the dependence of Λ_s on $\mathbf{y}_i^{(\ell)}$, we rely on an auxiliary quantity, namely $\|\mathbf{P}_{\|}(\mathbf{I} - \mathbf{Y}_{\Gamma}^{(\ell)}\mathbf{Y}_{\Gamma}^{(\ell)^\dagger})\mathbf{y}_i^{(\ell)}\|_2$ for a fixed index set $\Gamma \subset [n_\ell]\backslash\{i\}$ with cardinality satisfying $1 \leq |\Gamma| \leq \bar{s}$. We start the proof by deriving a lower bound φ_Γ on $\|\mathbf{P}_{\|}(\mathbf{I} - \mathbf{Y}_{\Gamma}^{(\ell)}\mathbf{Y}_{\Gamma}^{(\ell)^\dagger})\mathbf{y}_i^{(\ell)}\|_2$ and then show that $\mathcal{E}_7^{(\ell,i)} \supseteq \mathcal{F}_7^{(\ell,i)}$, where

$$\mathcal{F}_7^{(\ell,i)} := \left\{\varphi_{\Gamma'} > \left\|\mathbf{y}_{i\|}^{(\ell)}\right\|_2 \left(\frac{2}{3} - \sqrt{\frac{3\bar{s}\log((n_\ell - 1)e/\bar{s})}{d_\ell}}\right), \forall \Gamma' \in \mathcal{I}\right\}$$

with

$$\mathcal{I} := \{\Gamma' \subset [n_\ell]\backslash\{i\} \colon |\Gamma'| = \bar{s}\}, \qquad (2.40)$$

which implies $\mathrm{P}[\mathcal{E}_7^{(\ell,i)}] \geq \mathrm{P}[\mathcal{F}_7^{(\ell,i)}]$. The proof is then completed by establishing a lower bound on $\mathrm{P}[\mathcal{F}_7^{(\ell,i)}]$ using a version of Borell's inequality.

We proceed by lower-bounding $\|\mathbf{P}_{\|}(\mathbf{I} - \mathbf{Y}_{\Gamma}^{(\ell)}\mathbf{Y}_{\Gamma}^{(\ell)^\dagger})\mathbf{y}_i^{(\ell)}\|_2$ for $1 \leq |\Gamma| \leq \bar{s}$. Define the orthogonal projection matrices $\mathbf{P}_\Gamma := \mathbf{Y}_{\Gamma}^{(\ell)}\mathbf{Y}_{\Gamma}^{(\ell)^\dagger}$ and $\mathbf{P}_\Gamma^{\|} := \mathbf{Y}_{\Gamma\|}^{(\ell)}\mathbf{Y}_{\Gamma\|}^{(\ell)^\dagger}$ and note that $\mathbf{P}_{\|}\mathbf{P}_\Gamma^{\|} = \mathbf{P}_\Gamma^{\|} = \mathbf{P}_\Gamma^{\|}\mathbf{P}_{\|}$. We now get

$$\left\|\mathbf{P}_{\|}(\mathbf{I} - \mathbf{P}_\Gamma)\mathbf{y}_i^{(\ell)}\right\|_2^2 = \left\|\mathbf{P}_{\|}(\mathbf{I} - \mathbf{P}_\Gamma^{\|} + \mathbf{P}_\Gamma^{\|} - \mathbf{P}_\Gamma)\mathbf{y}_i^{(\ell)}\right\|_2^2 \qquad (2.41)$$
$$= \left\|\mathbf{P}_{\|}(\mathbf{I} - \mathbf{P}_\Gamma^{\|})\mathbf{y}_i^{(\ell)} + (\mathbf{P}_\Gamma^{\|} - \mathbf{P}_{\|}\mathbf{P}_\Gamma)\mathbf{y}_i^{(\ell)}\right\|_2^2 \qquad (2.42)$$

$$= \left\| \mathbf{P}_{\|}(\mathbf{I} - \mathbf{P}_{\Gamma}^{\|})\mathbf{y}_i^{(\ell)} \right\|_2^2 + \left\| (\mathbf{P}_{\Gamma}^{\|} - \mathbf{P}_{\|}\mathbf{P}_{\Gamma})\mathbf{y}_i^{(\ell)} \right\|_2^2 \tag{2.43}$$

$$\geq \left\| \mathbf{P}_{\|}(\mathbf{I} - \mathbf{P}_{\Gamma}^{\|})\mathbf{y}_i^{(\ell)} \right\|_2^2$$

$$= \left\| \mathbf{y}_{i\|}^{(\ell)} - \mathbf{P}_{\Gamma}^{\|}\mathbf{y}_{i\|}^{(\ell)} \right\|_2^2, \tag{2.44}$$

where the last equality is thanks to $\mathbf{P}_{\|}\mathbf{P}_{\Gamma}^{\|} = \mathbf{P}_{\Gamma}^{\|}\mathbf{P}_{\|}$ and the step leading from (2.42) to (2.43) follows from

$$\begin{aligned}
\left(\mathbf{P}_{\|}(\mathbf{I} - \mathbf{P}_{\Gamma}^{\|})\right)^{\top} (\mathbf{P}_{\Gamma}^{\|} - \mathbf{P}_{\|}\mathbf{P}_{\Gamma}) &= (\mathbf{I} - \mathbf{P}_{\Gamma}^{\|})\mathbf{P}_{\|}(\mathbf{P}_{\Gamma}^{\|} - \mathbf{P}_{\|}\mathbf{P}_{\Gamma}) \\
&= (\mathbf{I} - \mathbf{P}_{\Gamma}^{\|})\mathbf{P}_{\|}(\mathbf{P}_{\Gamma}^{\|}\mathbf{P}_{\Gamma}^{\|} - \mathbf{P}_{\Gamma}^{\|}\mathbf{P}_{\|}\mathbf{P}_{\Gamma}) \\
&= (\mathbf{I} - \mathbf{P}_{\Gamma}^{\|})\mathbf{P}_{\|}\mathbf{P}_{\Gamma}^{\|}(\mathbf{P}_{\Gamma}^{\|} - \mathbf{P}_{\|}\mathbf{P}_{\Gamma}) \\
&= \underbrace{(\mathbf{I} - \mathbf{P}_{\Gamma}^{\|})\mathbf{P}_{\Gamma}^{\|}}_{=\mathbf{0}}(\mathbf{P}_{\Gamma}^{\|} - \mathbf{P}_{\|}\mathbf{P}_{\Gamma}) = \mathbf{0}.
\end{aligned}$$

Using (2.41)–(2.44), we further have

$$\left\| \mathbf{P}_{\|}(\mathbf{I} - \mathbf{P}_{\Gamma})\mathbf{y}_i^{(\ell)} \right\|_2 \geq \left\| \mathbf{y}_{i\|}^{(\ell)} - \mathbf{P}_{\Gamma}^{\|}\mathbf{y}_{i\|}^{(\ell)} \right\|_2 \tag{2.45}$$

$$\geq \left\| \mathbf{y}_{i\|}^{(\ell)} \right\|_2 - \left\| \mathbf{P}_{\Gamma}^{\|}\mathbf{y}_{i\|}^{(\ell)} \right\|_2$$

$$\geq \underbrace{\left\| \mathbf{y}_{i\|}^{(\ell)} \right\|_2 - \left\| \mathbf{P}_{\Gamma'}^{\|}\mathbf{y}_{i\|}^{(\ell)} \right\|_2}_{=:\varphi_{\Gamma'}}, \tag{2.46}$$

for all $\Gamma \subseteq \Gamma' \in \mathcal{I}$, where the second inequality is by the reverse triangle inequality and the third is a consequence of $\mathcal{R}(\mathbf{P}_{\Gamma}^{\|}) \subseteq \mathcal{R}(\mathbf{P}_{\Gamma'}^{\|}) \subset \mathcal{S}_{\ell}$. It follows from (2.2) and (2.45)–(2.46) that $\|\mathbf{r}_{s\|}^{(\ell)}\|_2 = \|\mathbf{P}_{\|}(\mathbf{I} - \mathbf{P}_{\Lambda_s})\mathbf{y}_i^{(\ell)}\|_2 \geq \varphi_{\Gamma'}$ for $\Lambda_s \subset \mathcal{I}$, which implies $\|\mathbf{r}_{s\|}^{(\ell)}\|_2 \geq \min_{\Gamma' \in \mathcal{I}} \varphi_{\Gamma'}$, and thus, indeed, $\mathcal{E}_7^{(\ell,i)} \supseteq \mathcal{F}_7^{(\ell,i)}$. It remains to lower-bound $\mathrm{P}[\mathcal{F}_7^{(\ell,i)}]$.

To this end, we first work on the second term in (2.46) and note that

$$\begin{aligned}\frac{\left\|\mathbf{P}^{\|}_{\Gamma'}\mathbf{y}^{(\ell)}_{i\|}\right\|_2}{\left\|\mathbf{y}^{(\ell)}_{i\|}\right\|_2} &= \frac{\left\|\mathbf{Y}^{(\ell)}_{\Gamma'\|}\mathbf{Y}^{(\ell)\,\dagger}_{\Gamma'\|}\mathbf{y}^{(\ell)}_{i\|}\right\|_2}{\left\|\mathbf{y}^{(\ell)}_{i\|}\right\|_2}\\ &= \frac{\left\|\mathbf{U}^{(\ell)}\tilde{\mathbf{A}}^{(\ell)}_{\Gamma'}\left(\mathbf{U}^{(\ell)}\tilde{\mathbf{A}}^{(\ell)}_{\Gamma'}\right)^{\dagger}\mathbf{U}^{(\ell)}\tilde{\mathbf{a}}^{(\ell)}_i\right\|_2}{\left\|\mathbf{U}^{(\ell)}\tilde{\mathbf{a}}^{(\ell)}_i\right\|_2}\\ &= \frac{\left\|\tilde{\mathbf{A}}^{(\ell)}_{\Gamma'}\tilde{\mathbf{A}}^{(\ell)\dagger}_{\Gamma'}\tilde{\mathbf{a}}^{(\ell)}_i\right\|_2}{\left\|\tilde{\mathbf{a}}^{(\ell)}_i\right\|_2}. \end{aligned} \tag{2.47}$$

Since the columns of $\tilde{\mathbf{A}}^{(\ell)}_{\Gamma'}$, i.e., the vectors $\tilde{\mathbf{a}}^{(\ell)}_j, j \in \Gamma'$, are i.i.d. and of rotationally invariant distribution, $\tilde{\mathbf{A}}^{(\ell)}_{\Gamma'}\tilde{\mathbf{A}}^{(\ell)\dagger}_{\Gamma'}$ is the projector onto a subspace of $\mathbb{R}^{d_\ell}$ that is $\bar{s}$-dimensional w.p. 1. In particular, this subspace is distributed uniformly at random on the set of all $\bar{s}$-dimensional subspaces of $\mathbb{R}^{d_\ell}$. Indeed, note that we have, w.p. 1,

$$\begin{aligned}\mathcal{R}\left(\tilde{\mathbf{A}}^{(\ell)}_{\Gamma'}\right) &= \mathcal{R}\left(\tilde{\mathbf{A}}^{(\ell)}_{\Gamma'}\operatorname{diag}\left(1\Big/\left\|\tilde{\mathbf{a}}^{(\ell)}_{\gamma'_1}\right\|_2,\dots,1\Big/\left\|\tilde{\mathbf{a}}^{(\ell)}_{\gamma'_{\bar{s}}}\right\|_2\right)\right)\\ &\sim \mathcal{R}\left(\mathbf{G}\operatorname{diag}\left(1/\|\mathbf{g}_1\|_2,\dots,1/\|\mathbf{g}_{\bar{s}}\|_2\right)\right) && (2.48)\\ &= \mathcal{R}(\mathbf{G}) && (2.49)\\ &= \mathcal{R}(\mathbf{G}(\mathbf{G}^\top\mathbf{G})^{-1/2}), && (2.50)\end{aligned}$$

where γ'_j, $j \in [\bar{s}]$, denotes the elements of Γ' and $\mathbf{G} = [\mathbf{g}_1 \dots \mathbf{g}_{\bar{s}}] \in \mathbb{R}^{d_\ell \times \bar{s}}$ has i.i.d. standard normal random variables as entries. Here, to obtain (2.48), we used that the $\tilde{\mathbf{a}}^{(\ell)}_{\gamma'_j}/\|\tilde{\mathbf{a}}^{(\ell)}_{\gamma'_j}\|_2$, $j \in [\bar{s}]$, and the $\mathbf{g}_j/\|\mathbf{g}_j\|_2$, $j \in [\bar{s}]$, are all i.i.d. uniform on $\mathbb{S}^{d_\ell-1}$, and for (2.49) and (2.50) we exploit that $\operatorname{diag}(1/\|\mathbf{g}_1\|_2,\dots,1/\|\mathbf{g}_{\bar{s}}\|_2)$ and $\mathbf{G}$, respectively, have full rank w.p. 1. The claim now follows by noting that $\mathbf{G}(\mathbf{G}^\top\mathbf{G})^{-1/2}$ is distributed uniformly at random on the set of all orthonormal matrices in $\mathbb{R}^{d_\ell \times \bar{s}}$ (Chikuse, 2003, Thm. 2.2.1 iii)).

Next, we note that conditioning on $\tilde{\mathbf{A}}^{(\ell)}_{\Gamma'}\tilde{\mathbf{A}}^{(\ell)\dagger}_{\Gamma'}$ does not change the distribution of $\|\tilde{\mathbf{A}}_{\Gamma'}^{(\ell)}\tilde{\mathbf{A}}_{\Gamma'}^{(\ell)\dagger}\tilde{\mathbf{a}}_i^{(\ell)}\|_2^2/\|\tilde{\mathbf{a}}_i^{(\ell)}\|_2^2$. To see this, consider $\mathbf{a} \in \mathbb{R}^m$ distributed uniformly at random on $\mathbb{S}^{d_\ell - 1}$ and choose $\mathbf{V}$ uniformly at random from the set of all orthonormal matrices in $\mathbb{R}^{d_\ell \times \mathrm{d}_\ell}$. Further, let $\mathbf{P}_{\bar{s}}$ be a projector onto an arbitrary, but fixed $\bar{s}$-dimensional subspace of $\mathbb{R}^{d_\ell}$. Then, we have $\|\tilde{\mathbf{A}}_{\Gamma'}^{(\ell)}\tilde{\mathbf{A}}_{\Gamma'}^{(\ell)\dagger}\tilde{\mathbf{a}}_i^{(\ell)}\|_2^2/\|\tilde{\mathbf{a}}_i^{(\ell)}\|_2^2 \sim \left\|\mathbf{P}_{\bar{s}}\mathbf{V}^\top\mathbf{a}\right\|_2^2 \sim \left\|\mathbf{P}_{\bar{s}}\mathbf{a}\right\|_2^2$, where the first distributional equivalence follows from $\tilde{\mathbf{A}}_{\Gamma'}^{(\ell)}\tilde{\mathbf{A}}_{\Gamma'}^{(\ell)\dagger} \sim \mathbf{V}\mathbf{P}_{\bar{s}}\mathbf{V}^\top$ (by (Chikuse, 2003, Thm. 2.2.1 ii))) and the second from $\mathbf{V}^\top\mathbf{a} \sim \mathbf{a}$ (by rotational invariance of the distributions of $\mathbf{V}$ and $\mathbf{a}$).

Now, using $\|\mathbf{P}_{\Gamma'}^{\|}\mathbf{y}_{i\|}^{(\ell)}\|_2 = \|\tilde{\mathbf{A}}_{\Gamma'}^{(\ell)}\tilde{\mathbf{A}}_{\Gamma'}^{(\ell)\dagger}\tilde{\mathbf{a}}_i^{(\ell)}\|_2^2$ and conditioning $\|\tilde{\mathbf{A}}_{\Gamma'}^{(\ell)}\tilde{\mathbf{A}}_{\Gamma'}^{(\ell)\dagger}\tilde{\mathbf{a}}_i^{(\ell)}\|_2^2$ on $\tilde{\mathbf{A}}_{\Gamma'}^{(\ell)}\tilde{\mathbf{A}}_{\Gamma'}^{(\ell)\dagger}$ allows us to apply the following version of Borell's inequality to get an upper bound on the second term on the RHS of (2.46).

Lemma 2.6 (Extracted from the proof of (Soltanolkotabi and Candès, 2012, Lem. 7.5))**.** *Let* $\boldsymbol{\Sigma} \in \mathbb{R}^{d_1 \times d_2}$ *be a deterministic matrix and take* $\boldsymbol{\lambda} \in \mathbb{R}^{d_2}$ *to be distributed uniformly at random on* $\mathbb{S}^{d_2-1}$*. Then, we have*

$$\mathrm{P}\left[\|\boldsymbol{\Sigma}\boldsymbol{\lambda}\|_2 - \frac{\|\boldsymbol{\Sigma}\|_F}{\sqrt{d_2}} \geq \varepsilon\right] < e^{-d_2\varepsilon^2/(2\sigma_{\max}(\boldsymbol{\Sigma})^2)}.$$

Setting $\boldsymbol{\Sigma} = \tilde{\mathbf{A}}_{\Gamma'}^{(\ell)}\tilde{\mathbf{A}}_{\Gamma'}^{(\ell)\dagger}$ and $\boldsymbol{\lambda} = \tilde{\mathbf{a}}_i^{(\ell)}/\|\tilde{\mathbf{a}}_i^{(\ell)}\|_2$ in Lemma 2.6 and noting that $\|\boldsymbol{\Sigma}\|_F = \|\tilde{\mathbf{A}}_{\Gamma'}^{(\ell)}\tilde{\mathbf{A}}_{\Gamma'}^{(\ell)\dagger}\|_F = \sqrt{\bar{s}}$ and $\sigma_{\max}(\boldsymbol{\Sigma}) = 1$ yields

$$\mathrm{P}\left[\left\|\tilde{\mathbf{A}}_{\Gamma'}^{(\ell)}\tilde{\mathbf{A}}_{\Gamma'}^{(\ell)\dagger}\tilde{\mathbf{a}}_i^{(\ell)}\right\|_2 \geq \left\|\tilde{\mathbf{a}}_i^{(\ell)}\right\|_2\left(\sqrt{\frac{\bar{s}}{d_\ell}} + \varepsilon\right)\right] < e^{-d_\ell\varepsilon^2/2}. \tag{2.51}$$

We now have

$$\begin{aligned}
&\mathrm{P}\left[\varphi_{\Gamma'} \geq \left\|\mathbf{y}_{i\|}^{(\ell)}\right\|_2\left(1 - \sqrt{\frac{\bar{s}}{d_\ell}} - \varepsilon\right),\ \forall\Gamma' \in \mathcal{I}\right] \\
&\quad = \mathrm{P}\left[\left\|\mathbf{P}_{\|\Gamma'}\mathbf{y}_{i\|}^{(\ell)}\right\|_2 < \left\|\mathbf{y}_{i\|}^{(\ell)}\right\|_2\left(\sqrt{\frac{\bar{s}}{d_\ell}} + \varepsilon\right),\ \forall\Gamma' \in \mathcal{I}\right] && (2.52) \\
&\quad \geq 1 - \sum_{\Gamma' \in \mathcal{I}} \mathrm{P}\left[\left\|\mathbf{P}_{\|\Gamma'}\mathbf{y}_{i\|}^{(\ell)}\right\|_2 \geq \left\|\mathbf{y}_{i\|}^{(\ell)}\right\|_2\left(\sqrt{\frac{\bar{s}}{d_\ell}} + \varepsilon\right)\right] && (2.53)
\end{aligned}$$

$$\geq 1 - \binom{n_\ell - 1}{\bar{s}} e^{-d_\ell \varepsilon^2/2} \tag{2.54}$$

$$\geq 1 - e^{\bar{s}\log((n_\ell-1)e/\bar{s})} e^{-d_\ell \varepsilon^2/2}, \tag{2.55}$$

where we used the inequality (2.45)–(2.46) to get (2.52), a union bound for the step leading from (2.52) to (2.53), (2.51) and $|\mathcal{I}| = \binom{n_\ell - 1}{\bar{s}}$ to obtain (2.54) (recall that $\|\mathbf{P}_{\|\Gamma'}\mathbf{y}_{i\|}^{(\ell)}\|_2 = \|\tilde{\mathbf{A}}_{\Gamma'}^{(\ell)}\tilde{\mathbf{A}}_{\Gamma'}^{(\ell)\dagger}\tilde{\mathbf{a}}_i^{(\ell)}\|_2$ and $\|\mathbf{y}_i^{(\ell)}\|_2 = \|\tilde{\mathbf{a}}_i^{(\ell)}\|_2$), and $\binom{n_\ell - 1}{\bar{s}} \leq ((n_\ell - 1)e/\bar{s})^{\bar{s}}$ to get (2.55). Finally, setting

$$\varepsilon = \sqrt{\frac{1}{9} + \frac{2\bar{s}\log((n_\ell-1)e/\bar{s})}{d_\ell}} < \frac{1}{3} + \sqrt{\frac{2\bar{s}\log((n_\ell-1)e/\bar{s})}{d_\ell}}$$

in (2.55) and noting that

$$\begin{aligned} 1 - \sqrt{\frac{\bar{s}}{d_\ell}} - \varepsilon &> \frac{2}{3} - \sqrt{\frac{\bar{s}}{d_\ell}} - \sqrt{\frac{2\bar{s}\log((n_\ell-1)e/\bar{s})}{d_\ell}} \\ &> \frac{2}{3} - \sqrt{\frac{3\bar{s}\log((n_\ell-1)e/\bar{s})}{d_\ell}}, \end{aligned}$$

where we used $\log((n_\ell-1)e/\bar{s}) \geq 1$ (as $n_\ell - 1 \geq d_\ell$ and $\bar{s} \leq d_\ell$) for the last inequality, we have

$$\begin{aligned} \mathrm{P}\left[\mathcal{F}_7^{(\ell,i)}\right] &= \mathrm{P}\left[\varphi_{\Gamma'} > \left\|\mathbf{y}_{i\|}^{(\ell)}\right\|_2 \left(\frac{2}{3} - \sqrt{\frac{3\bar{s}\log((n_\ell-1)e/\bar{s})}{d_\ell}}\right), \forall \Gamma' \in \mathcal{I}\right] \\ &\geq 1 - e^{-d_\ell/18}. \end{aligned}$$

This completes the proof of Lemma 2.5. □

We continue by deriving a lower bound on $\mathrm{P}[\mathcal{E}_6^{(\ell,i)}]$.

Lemma 2.7. *Set* $\tilde{a} := \min_{j \in [n_\ell]\setminus\{i\}} \|\mathbf{y}_{j\|}^{(\ell)}\|_2$ *and suppose that* $s_{\max} \leq c_1 d_\ell / \log(e(n_\ell - 1)/s_{\max})$ *for a numerical constant* $c_1 > 0$*. Then, we* *have*

$$\mathrm{P}\left[\mathcal{E}_6^{(\ell,i)}\right] = \mathrm{P}\left[\left\|\mathbf{r}_{s\perp}^{(\ell)}\right\|_2 \leq \left\|\mathbf{z}_{i\perp}^{(\ell)}\right\|_2 + \frac{3\sigma}{\tilde{a}}\left\|\mathbf{y}_i^{(\ell)}\right\|_2, \ \forall s \leq s_{\max}\right]$$
$$\geq 1 - 2e^{-c_2 m} - 2e^{-c_3 d_\ell}, \tag{2.56}$$

where $c_2, c_3 > 0$ *are numerical constants.*

Proof. First note that for $s = 0$, $\mathbf{r}_{0\perp}^{(\ell)} = \mathbf{y}_{i\perp}^{(\ell)} = \mathbf{z}_{i\perp}^{(\ell)}$ and the inequality $\|\mathbf{r}_{s\perp}^{(\ell)}\|_2 \leq \|\mathbf{z}_{i\perp}^{(\ell)}\|_2 + (3\sigma/\tilde{a})\|\mathbf{y}_i^{(\ell)}\|_2$ holds trivially. For $1 \leq s \leq s_{\max}$, as in the proof of Lemma 2.5, we consider fixed index sets $\Gamma \in \mathcal{J}$, with

$$\mathcal{J} \coloneqq \{\Gamma' \subset [n_\ell]\backslash\{i\} \colon |\Gamma'| = s \in [s_{\max}]\}, \tag{2.57}$$

to resolve the issue of statistical dependencies (between the columns of $\mathbf{Y}_{\Lambda_s}^{(\ell)}$ and $\mathbf{y}_i^{(\ell)}$) in $\|\mathbf{r}_{s\perp}^{(\ell)}\|_2 = \|\mathbf{P}_\perp(\mathbf{I} - \mathbf{Y}_{\Lambda_s}^{(\ell)}\mathbf{Y}_{\Lambda_s}^{(\ell)\dagger})\mathbf{y}_i^{(\ell)}\|_2$. Specifically, this is accomplished by upper-bounding $\|\mathbf{P}_\perp(\mathbf{I} - \mathbf{Y}_\Gamma^{(\ell)}\mathbf{Y}_\Gamma^{(\ell)\dagger})\mathbf{y}_i^{(\ell)}\|_2$ according to (2.58) and establishing that the submatrix $\mathbf{Y}_\Gamma^{(\ell)}$ of $\mathbf{Y}^{(\ell)}$ is well-conditioned with high probability for all $\Gamma \in \mathcal{J}$, in particular for the sets $\Lambda_s \in \mathcal{J}$, $s \in [s_{\max}]$, which determine $\mathbf{r}_{s\perp}^{(\ell)}$. This will be accomplished by employing the restricted isometry property (RIP) (Vershynin, 2012, Sec. 5.6) and standard concentration of measure results from random matrix theory.

By the triangle inequality and the submultiplicativity of the operator norm, we have, for every $\Gamma \in \mathcal{J}$, that

$$\left\|\mathbf{P}_\perp(\mathbf{I} - \mathbf{Y}_\Gamma^{(\ell)}\mathbf{Y}_\Gamma^{(\ell)\dagger})\mathbf{y}_i^{(\ell)}\right\|_2$$
$$\leq \left\|\mathbf{y}_{i\perp}^{(\ell)}\right\|_2 + \left\|\mathbf{P}_\perp\mathbf{Y}_\Gamma^{(\ell)}\right\|_{2\to2}\left\|\mathbf{Y}_\Gamma^{(\ell)\dagger}\right\|_{2\to2}\left\|\mathbf{y}_i^{(\ell)}\right\|_2 \tag{2.58}$$
$$= \left\|\mathbf{z}_{i\perp}^{(\ell)}\right\|_2 + \frac{1}{\sigma_{\min}(\mathbf{Y}_\Gamma^{(\ell)})}\left\|\mathbf{P}_\perp\mathbf{Z}_\Gamma^{(\ell)}\right\|_{2\to2}\left\|\mathbf{y}_i^{(\ell)}\right\|_2 \tag{2.59}$$
$$\leq \left\|\mathbf{z}_{i\perp}^{(\ell)}\right\|_2 + \frac{1}{\sigma_{\min}(\tilde{\mathbf{A}}_\Gamma^{(\ell)})}\left\|\mathbf{Z}_\Gamma^{(\ell)}\right\|_{2\to2}\left\|\mathbf{y}_i^{(\ell)}\right\|_2, \tag{2.60}$$

where we used $\sigma_{\min}(\mathbf{Y}_\Gamma^{(\ell)}) = 1/\|\mathbf{Y}_\Gamma^{(\ell)\dagger}\|_{2\to 2}$ (Vershynin, 2012, Sec. 5.2.1) to get (2.59) and $\sigma_{\min}(\mathbf{Y}_\Gamma^{(\ell)}) \geq \sigma_{\min}(\tilde{\mathbf{A}}_\Gamma^{(\ell)}) > 0$ w.p. 1 (where the first inequality is a consequence of $\|\mathbf{Y}_\Gamma^{(\ell)}\mathbf{v}\|_2 \geq \|\mathbf{P}_\| \mathbf{Y}_\Gamma^{(\ell)}\mathbf{v}\|_2 = \|\tilde{\mathbf{A}}_\Gamma^{(\ell)}\mathbf{v}\|_2$, for all $\mathbf{v} \in \mathbb{R}^s$, and the second stems from the fact that $\tilde{\mathbf{A}}_\Gamma^{(\ell)}$ has full column rank w.p. 1) to get (2.60). Denote the elements of Γ by γ_j, $j \in [s]$. We continue by decomposing $\tilde{\mathbf{A}}_\Gamma^{(\ell)}$ according to $\tilde{\mathbf{A}}_\Gamma^{(\ell)} = \mathbf{E}_\Gamma \mathbf{D}_\Gamma$, where $\mathbf{E} := [\tilde{\mathbf{a}}_1^{(\ell)}/\|\tilde{\mathbf{a}}_1^{(\ell)}\|_2 \; \tilde{\mathbf{a}}_2^{(\ell)}/\|\tilde{\mathbf{a}}_2^{(\ell)}\|_2 \; \ldots \; \tilde{\mathbf{a}}_{n_\ell}^{(\ell)}/\|\tilde{\mathbf{a}}_{n_\ell}^{(\ell)}\|_2]$, and $\mathbf{D}_\Gamma := \operatorname{diag}(\|\tilde{\mathbf{a}}_{\gamma_1}^{(\ell)}\|_2, \|\tilde{\mathbf{a}}_{\gamma_2}^{(\ell)}\|_2, \ldots, \|\tilde{\mathbf{a}}_{\gamma_s}^{(\ell)}\|_2)$. Note that the columns of $\mathbf{E}$ are distributed i.i.d. uniformly at random on $\mathbb{S}^{d_\ell - 1}$ and $\sigma_{\min}(\mathbf{D}_\Gamma) \geq \tilde{a}$. We next establish that $\tilde{\mathbf{A}}^{(\ell)}$ and $\mathbf{Z}^{(\ell)}$ satisfy the RIP with high probability, which will then allow us to bound $\sigma_{\min}(\tilde{\mathbf{A}}_\Gamma^{(\ell)})$ and $\sigma_{\max}(\mathbf{Z}_\Gamma^{(\ell)})$, respectively, in (2.60), for all $\Gamma \in \mathcal{J}$.

We start by recalling the definition of the RIP.

Definition 2.2. *A matrix* $\mathbf{A} \in \mathbb{R}^{m \times n}$ *satisfies the RIP of order* $p \geq 1$ *if there exists* $\delta_p > 0$ *such that*

$$(1 - \delta_p)\|\mathbf{v}\|_2^2 \leq \|\mathbf{A}\mathbf{v}\|_2^2 \leq (1 + \delta_p)\|\mathbf{v}\|_2^2 \tag{2.61}$$

holds for all $\mathbf{v} \in \mathbb{R}^n$ *with* $\|\mathbf{v}\|_0 \leq p$. *The smallest number* $\delta_p = \delta_p(\mathbf{A})$ *satisfying* (2.61) *is called the restricted isometry constant of* $\mathbf{A}$.

If $\mathbf{A} \in \mathbb{R}^{m \times n}$ satisfies the RIP of order p, it follows from (Vershynin, 2012, Lem. 5.36) that for $\delta \in [\delta_p, 1]$,

$$1 - \delta \leq \sigma_{\min}(\mathbf{A}_\mathcal{T}) \leq \sigma_{\max}(\mathbf{A}_\mathcal{T}) \leq 1 + \delta, \qquad \text{for all } \mathcal{T} \subseteq [n] \text{ with } |\mathcal{T}| \leq p. \tag{2.62}$$

By (Vershynin, 2012, Ex. 5.25) the rows of $(\sqrt{m}/\sigma)\mathbf{Z}_{-i}^{(\ell)} \in \mathbb{R}^{m \times (n_\ell - 1)}$ are independent sub-gaussian isotropic random vectors (Vershynin, 2012, Def. 5.19, Def. 5.22), and by (Vershynin, 2012, Ex. 5.25) the columns of $\sqrt{d_\ell}\mathbf{E}_{-i} \in \mathbb{R}^{d_\ell \times (n_\ell - 1)}$ are independent sub-gaussian isotropic random vectors with ℓ_2-norm $\sqrt{d_\ell}$ a.s. We can therefore apply the next lemma to show that $(\sqrt{m}/\sigma)\mathbf{Z}_{-i}^{(\ell)}$ and $\sqrt{d_\ell}\mathbf{E}_{-i}$ satisfy the RIP for suitable p and δ with high probability. This will then allow us to bound $\sigma_{\min}(\tilde{\mathbf{A}}_\Gamma^{(\ell)})$ and $\sigma_{\max}(\mathbf{Z}_\Gamma^{(\ell)})$ for all $\Gamma \in \mathcal{J}$.

Lemma 2.8 (Vershynin (2012), Thm. 5.65). *Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be a random matrix with independent sub-gaussian isotropic random vectors as rows or independent sub-gaussian isotropic random vectors with ℓ_2-norm $\sqrt{m}$ a.s. as columns. Select p with $1 \leq p \leq n$ and let $\delta \in (0, 1)$. If*

$$m \geq C\delta^{-2} p \log(en/p),$$

then, w.p. at least $1 - 2e^{-c\delta^2 m}$, the normalized matrix $\bar{\mathbf{A}} := (1/\sqrt{m})\mathbf{A}$ satisfies the RIP of order p with $\delta_p(\bar{\mathbf{A}}) \leq \delta$. Here, the constants $c, C > 0$ depend only on the sub-gaussian norm[4] *of the rows or columns of $\mathbf{A}$.*

By Lemma 2.8 with $\mathbf{A} = (\sqrt{m}/\sigma)\mathbf{Z}_{-i}^{(\ell)}$ and $\delta = 1/2$, and (2.62), if $s_{\max} \leq C_2 m / \log(e(n_\ell - 1)/s_{\max})$, there exist constants $c_2, C_2 > 0$ such that

$$\mathrm{P}\left[\frac{1}{2}\sigma \leq \sigma_{\min}(\mathbf{Z}_\Gamma^{(\ell)}) \leq \sigma_{\max}(\mathbf{Z}_\Gamma^{(\ell)}) \leq \frac{3}{2}\sigma, \ \ \forall \Gamma \in \mathcal{J}\right] \geq 1 - 2e^{-c_2 m}. \tag{2.63}$$

Setting $\mathbf{A} = \sqrt{d_\ell}\mathbf{E}_{-i}$ and $\delta = 1/2$ in Lemma 2.8, we have similarly

$$\mathrm{P}\left[\frac{1}{2} \leq \sigma_{\min}(\mathbf{E}_\Gamma) \leq \sigma_{\max}(\mathbf{E}_\Gamma) \leq \frac{3}{2}, \ \forall \Gamma \in \mathcal{J}\right] \geq 1 - 2e^{-c_3 d_\ell}, \tag{2.64}$$

if $s_{\max} \leq C_3 d_\ell / \log(e(n_\ell - 1)/s_{\max})$, for constants $c_3, C_3 > 0$. Putting (2.63) and (2.64) together, we get (2.56) as follows

$$\begin{aligned}\mathrm{P}\left[\mathcal{E}_6^{(\ell,i)}\right] &= \mathrm{P}\left[\left\|\mathbf{P}_\perp(\mathbf{I} - \mathbf{Y}_{\Lambda_s}^{(\ell)}\mathbf{Y}_{\Lambda_s}^{(\ell)\dagger})\mathbf{y}_i^{(\ell)}\right\|_2 \leq \left\|\mathbf{z}_{i\perp}^{(\ell)}\right\|_2 + \frac{3\sigma}{\tilde{a}}\left\|\mathbf{y}_i^{(\ell)}\right\|_2\right] \\ &\geq \mathrm{P}\left[\frac{\sigma_{\max}(\mathbf{Z}_{\Lambda_s}^{(\ell)})}{\sigma_{\min}(\tilde{\mathbf{A}}_{\Lambda_s}^{(\ell)})} \leq \frac{3\sigma}{\tilde{a}}\right] \end{aligned} \tag{2.65}$$

[4] The sub-gaussian norm of a random variable X is defined as $\|X\|_{\Psi_2} := \sup_{p \geq 1} p^{-1/2}(\mathbb{E}[|X|^p])^{1/p}$ (Vershynin, 2012, Def. 5.7).

$$\begin{aligned}
&\geq \mathrm{P}\left[\frac{\sigma_{\max}(\mathbf{Z}^{(\ell)}_{\Lambda_s})}{\sigma_{\min}(\mathbf{E}_{\Lambda_s})} \leq 3\sigma\right] && (2.66)\\
&\geq \mathrm{P}\left[\left\{\sigma_{\min}(\mathbf{E}_{\Lambda_s}) \geq \frac{1}{2}\right\} \cap \left\{\sigma_{\max}(\mathbf{Z}^{(\ell)}_{\Lambda_s}) \leq \frac{3}{2}\sigma\right\}\right]\\
&\geq 1 - \mathrm{P}\left[\sigma_{\min}(\mathbf{E}_{\Lambda_s}) < \frac{1}{2}\right] - \mathrm{P}\left[\sigma_{\max}(\mathbf{Z}^{(\ell)}_{\Lambda_s}) > \frac{3}{2}\sigma\right] && (2.67)\\
&\geq 1 - 2e^{-c_2 m} - 2e^{-c_3 d_\ell}, && (2.68)
\end{aligned}$$

for all $s \leq s_{\max}$. Here (2.65) follows from (2.60) with $\Gamma = \Lambda_s$, (2.66) is by

$$\frac{1}{\sigma_{\min}(\tilde{\mathbf{A}}^{(\ell)}_{\Lambda_s})} \leq \frac{1}{\tilde{a}\sigma_{\min}(\mathbf{E}_{\Lambda_s})},$$

where we used

$$\begin{aligned}
\sigma_{\min}(\tilde{\mathbf{A}}^{(\ell)}_{\Lambda_s}) &= \min_{\mathbf{v}\in\mathbb{R}^s} \|\mathbf{E}_{\Lambda_s}\mathbf{D}_{\Lambda_s}\mathbf{v}\|_2\\
&= \min_{\mathbf{v}\in\mathbb{R}^s} \left\|\mathbf{E}_{\Lambda_s}\frac{\mathbf{D}_{\Lambda_s}\mathbf{v}}{\|\mathbf{D}_{\Lambda_s}\mathbf{v}\|_2}\right\|_2 \|\mathbf{D}_{\Lambda_s}\mathbf{v}\|_2\\
&\geq \min_{\mathbf{v}\in\mathbb{R}^s} \left\|\mathbf{E}_{\Lambda_s}\frac{\mathbf{D}_{\Lambda_s}\mathbf{v}}{\|\mathbf{D}_{\Lambda_s}\mathbf{v}\|_2}\right\|_2 \tilde{a}\\
&= \tilde{a}\sigma_{\min}(\mathbf{E}_{\Lambda_s}),
\end{aligned}$$

(2.67) is by a union bound, and (2.68) follows from (2.63) and (2.64) with $\Lambda_s \in \mathcal{J}$ for all $s \leq s_{\max}$. Finally, letting $c_1 := \min\{2C_2, C_3\} \leq \min\{(m/d_\ell)C_2, C_3\}$ (using the assumption $m \geq 2d_{\max}$) ensures that $s_{\max}$, and thereby $|\Gamma|$, is small enough for both (2.63) and (2.64) to hold. This concludes the proof of Lemma 2.7. □

We proceed by establishing a lower bound on $\mathrm{P}[\mathcal{E}_5^{(\ell,i)}]$.

Lemma 2.9. *For numerical constants $c_4, c_5 > 0$ it holds that*

$$\mathrm{P}\left[\mathcal{E}_5^{(\ell,i)}\right] = \mathrm{P}\left[\max_{j\in[n_\ell]\setminus(\Lambda_s\cup\{i\})} \left|\left\langle \mathbf{y}_j^{(\ell)}, \mathbf{r}_s^{(\ell)}\right\rangle\right| \geq \left(1 - \frac{c_4+1}{\sqrt{\rho_\ell}}\right)\frac{\left\|\mathbf{r}_{s\|}^{(\ell)}\right\|_2}{\sqrt{d_\ell}}\right.$$

$$-\sigma\left(\frac{1}{\sqrt{m}}+\frac{2}{\sqrt{n_\ell-1}}\right)\left\|\mathbf{r}_s^{(\ell)}\right\|_2\Bigg]$$

$$\geq 1-2e^{-c_5 d_\ell}-2e^{-m/2}. \tag{2.69}$$

Proof. We first lower-bound the maximum in (2.69) by a term proportional to $\|\mathbf{Y}_{-i}^{(\ell)^\top}\mathbf{r}_s^{(\ell)}\|_2$ and then establish (2.69) by leveraging standard bounds on the singular values of random matrices. We have

$$\begin{aligned}
\max_{j\in[n_\ell]\setminus(\Lambda_s\cup\{i\})}\left|\left\langle \mathbf{y}_j^{(\ell)},\mathbf{r}_s^{(\ell)}\right\rangle\right| &= \max_{j\in[n_\ell]\setminus\{i\}}\left|\left\langle \mathbf{y}_j^{(\ell)},\mathbf{r}_s^{(\ell)}\right\rangle\right| \\
&= \left\|\mathbf{Y}_{-i}^{(\ell)^\top}\mathbf{r}_s^{(\ell)}\right\|_\infty \\
&\geq \frac{\left\|\mathbf{Y}_{-i}^{(\ell)^\top}\mathbf{r}_s^{(\ell)}\right\|_2}{\sqrt{n_\ell-1}} \\
&= \frac{\left\|\mathbf{A}_{-i}^{(\ell)^\top}\mathbf{U}^{(\ell)^\top}\mathbf{r}_{s\|}^{(\ell)}+\mathbf{Z}_{-i}^{(\ell)^\top}\mathbf{r}_s^{(\ell)}\right\|_2}{\sqrt{n_\ell-1}} \\
&\geq \frac{\left\|\mathbf{A}_{-i}^{(\ell)^\top}\mathbf{U}^{(\ell)^\top}\mathbf{r}_{s\|}^{(\ell)}\right\|_2}{\sqrt{n_\ell-1}}-\frac{\left\|\mathbf{Z}_{-i}^{(\ell)^\top}\mathbf{r}_s^{(\ell)}\right\|_2}{\sqrt{n_\ell-1}} \\
&\geq \frac{\sigma_{\min}\left(\mathbf{A}_{-i}^{(\ell)^\top}\right)}{\sqrt{n_\ell-1}}\left\|\mathbf{r}_{s\|}^{(\ell)}\right\|_2-\frac{\sigma_{\max}\left(\mathbf{Z}_{-i}^{(\ell)^\top}\right)}{\sqrt{n_\ell-1}}\left\|\mathbf{r}_s^{(\ell)}\right\|_2,
\end{aligned} \tag{2.70}$$

where the first equality is thanks to orthogonality of $\mathbf{r}_s^{(\ell)}$ and $\mathbf{y}_j^{(\ell)}$, for all $j\in\Lambda_s$, the first inequality follows from $\|\mathbf{v}\|_2\leq\sqrt{n_\ell-1}\|\mathbf{v}\|_\infty$, for all $\mathbf{v}\in\mathbb{R}^{n_\ell-1}$, and the second inequality is by the reverse triangle inequality.

Noting that $\sqrt{d_\ell}\mathbf{A}_{-i}^{(\ell)^\top}$ is a $(n_\ell-1)\times d_\ell$ matrix whose rows are independent isotropic subgaussian random vectors (as defined in (Vershynin, 2012, Def. 5.19, Def. 5.22)), it follows from (Vershynin, 2012, Thm. 5.39) (see Theorem 2.5 in Appendix 2.D) that

$$\mathrm{P}\left[\sqrt{d_\ell}\sigma_{\min}\left({\mathbf{A}_{-i}^{(\ell)}}^\top\right) < \sqrt{n_\ell - 1} - c_4\sqrt{d_\ell} - t\right] < 2e^{-c_5 t^2}, \quad (2.71)$$

where the constants $c_4, c_5 > 0$ depend only on the sub-gaussian norm of the rows of ${\mathbf{A}_{-i}^{(\ell)}}^\top$. Setting $t = \sqrt{d_\ell}$ in (2.71), we get

$$\mathrm{P}\left[\frac{\sigma_{\min}\left({\mathbf{A}_{-i}^{(\ell)}}^\top\right)}{\sqrt{n_\ell - 1}}\left\|\mathbf{r}_{s\|}^{(\ell)}\right\|_2 < \left(1 - \frac{c_4 + 1}{\sqrt{\rho_\ell}}\right)\frac{\left\|\mathbf{r}_{s\|}^{(\ell)}\right\|_2}{\sqrt{d_\ell}}\right] < 2e^{-c_5 d_\ell}. \quad (2.72)$$

Since $(\sqrt{m}/\sigma){\mathbf{Z}_{-i}^{(\ell)}}^\top$ is a $(n_\ell - 1) \times m$ matrix with i.i.d. standard normal entries, it follows from (Vershynin, 2012, Cor. 5.35) (see Corollary 2.6 in Appendix 2.D) that

$$\mathrm{P}\left[\frac{\sqrt{m}}{\sigma}\sigma_{\max}\left({\mathbf{Z}_{-i}^{(\ell)}}^\top\right) > \sqrt{n_\ell - 1} + \sqrt{m} + t\right] < 2e^{-t^2/2}. \quad (2.73)$$

Setting $t = \sqrt{m}$ in (2.73), we obtain

$$\mathrm{P}\left[\frac{\sigma_{\max}\left({\mathbf{Z}_{-i}^{(\ell)}}^\top\right)}{\sqrt{n_\ell - 1}}\left\|\mathbf{r}_{s}^{(\ell)}\right\|_2 > \sigma\left(\frac{1}{\sqrt{m}} + \frac{2}{\sqrt{n_\ell - 1}}\right)\left\|\mathbf{r}_{s}^{(\ell)}\right\|_2\right] < 2e^{-m/2}. \quad (2.74)$$

The claim in Lemma 2.9 now follows by lower-bounding the first term in (2.70) using (2.72), by upper-bounding the second term in (2.70) using (2.74), and by application of a union bound. □

Finally, we derive a lower bound on $\mathrm{P}[\mathcal{E}_4]$.

Lemma 2.10. *We have*

$$\mathrm{P}[\mathcal{E}_4] \geq 1 - \sum_{\ell \in [L]} n_\ell (e^{-d_\ell/8} + e^{-m/8}). \tag{2.75}$$

Proof. The proof is effected by applying the following well-known concentration result.

Theorem 2.4 (Ledoux (2005))**.** *Let $f\colon \mathbb{R}^m \to \mathbb{R}$ be a Lipschitz function with Lipschitz constant K, i.e., $|f(\mathbf{a}) - f(\mathbf{b})| \leq K\|\mathbf{a} - \mathbf{b}\|_2$, for all $\mathbf{a}, \mathbf{b} \in \mathbb{R}^m$. Let $\mathbf{z} \in \mathbb{R}^m$ be a $\mathcal{N}(\mathbf{0}, \mathbf{I}_m)$ vector. Then, for $t \geq 0$, we have*

$$\mathrm{P}[f(\mathbf{z}) - \mathbb{E}[f(\mathbf{z})] > t] \leq e^{-t^2/(2K^2)}. \tag{2.76}$$

The functions $f(\mathbf{z}) = \|\mathbf{z}\|_2$ and $f_{\|}(\mathbf{z}) = \|\mathbf{P}_{\|}\mathbf{z}\|_2$ both have Lipschitz constant $K = 1$ ($|f_{\|}(\mathbf{a}) - f_{\|}(\mathbf{b})| = |\|\mathbf{P}_{\|}\mathbf{a}\|_2 - \|\mathbf{P}_{\|}\mathbf{b}\|_2| \leq \|\mathbf{P}_{\|}(\mathbf{a} - \mathbf{b})\|_2 \leq \|\mathbf{a} - \mathbf{b}\|_2$, for all $\mathbf{a}, \mathbf{b} \in \mathbb{R}^m$, by the reverse triangle inequality and $\|\mathbf{P}_{\|}\|_{2\to 2} = 1$). For $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_m)$, we get by Jensen's inequality $\mathbb{E}[\|\mathbf{z}\|_2] \leq \sqrt{\mathbb{E}[\|\mathbf{z}\|_2^2]} = \sqrt{m}$ and $\mathbb{E}[\|\mathbf{P}_{\|}\mathbf{z}\|_2] \leq \sqrt{\mathbb{E}[\|\mathbf{P}_{\|}\mathbf{z}\|_2^2]} = \sqrt{d_\ell}$ (where the equality follows from $\mathbf{P}_{\|}\mathbf{z} = \mathbf{U}^{(\ell)}\mathbf{U}^{(\ell)\top}\mathbf{z}$ and the fact that $\mathbf{U}^{(\ell)\top}\mathbf{z}$ is $\mathcal{N}(0, \mathbf{I}_{d_\ell})$-distributed). Noting that $\mathbf{z}_i^{(\ell)} \sim (\sigma/\sqrt{m})\mathbf{z}$, application of (2.76) to $\mathbf{z}$ and $\mathbf{P}_{\|}\mathbf{z}$ yields

$$\mathrm{P}\left[\left\|\mathbf{z}_i^{(\ell)}\right\|_2 > \frac{3}{2}\sigma\right] \leq e^{-m/8} \qquad \text{and}$$
$$\mathrm{P}\left[\left\|\mathbf{P}_{\|}\mathbf{z}_i^{(\ell)}\right\|_2 > \frac{3\sqrt{d_\ell}}{2\sqrt{m}}\sigma\right] \leq e^{-d_\ell/8},$$

where we set $t = \sqrt{m}/2$ and $t = \sqrt{d_\ell}/2$, respectively. A union bound over $\ell \in [L]$, $i \in [n_\ell]$, yields the desired lower bound (2.75). □

2.B. PROOF OF THEOREM 2.2

Most steps of the proof of Theorem 2.2 are almost identical to those in the proof of Theorem 2.1. We therefore elaborate only on the arguments the proofs differ in significantly.

Analogously to the proof for SSC-OMP, we will henceforth work with the "reduced MP" algorithm which, for the representation of $\mathbf{y}_i^{(\ell)}$ selects elements from the reduced dictionary $\mathcal{Y}_\ell \backslash \{\mathbf{y}_i^{(\ell)}\}$ only, instead of the full dictionary $\mathcal{Y}\backslash\{\mathbf{y}_i^{(\ell)}\}$. The justification for relying on reduced MP to establish the desired result is identical to that for OMP. Throughout the proof the residual of the reduced MP algorithm will be denoted by $\mathbf{q}_s^{(\ell)}$ and the number of iterations actually performed when a stopping condition has been met by s_a. As in the OMP case, for expositional convenience, the quantities $\mathbf{q}_s^{(\ell)}$ and s_a do not reflect the dependence on the index i of the data point $\mathbf{y}_i^{(\ell)}$.

Note that the selection rules for OMP and MP are equivalent in the following sense. As the OMP residual $\mathbf{r}_s^{(\ell)}$ is orthogonal to $\mathbf{y}_j^{(\ell)}$, for all $j \in \Lambda_s$, we can replace $\max_{j\in[N]\backslash(\Lambda_s\cup\{i\})} |\langle \mathbf{y}_j^{(\ell)}, \mathbf{r}_s\rangle|$ on the RHS of (2.16) by $\max_{j\in[N]\backslash\{i\}} |\langle \mathbf{y}_j^{(\ell)}, \mathbf{r}_s\rangle|$, i.e., we can take the maximization over $j \in [N]\backslash\{i\}$ as in MP. We therefore need to show that (2.16) with $\mathbf{r}_s$ replaced by $\mathbf{q}_s$ and $\max_{j\in[N]\backslash(\Lambda_s\cup\{i\})}$ replaced by $\max_{j\in[N]\backslash\{i\}}$ holds for all MP iterations, and for every $\mathbf{y}_i^{(\ell)} \in \mathcal{Y}_\ell$, $\ell \in [L]$, w.p. at least $P^\star$.

Next, we systematically revisit the events $\mathcal{E}_1^{(\ell,i,s)}$– $\mathcal{E}_7^{(\ell,i)}$ and adapt the corresponding bounds for MP where needed. Recall that the bounds on $\mathrm{P}[\mathcal{E}_1^{(\ell,i,s)}]$ and $\mathrm{P}[\mathcal{E}_2^{(\ell,i,s)}]$ in Lemma 2.2 rely on the rotational invariance—as expressed in Lemma 2.1—of the distributions of $\mathbf{r}_{s\|}^{(\ell)}$ and $\mathbf{r}_{s\perp}^{(\ell)}$, respectively. As the residual update rule (2.5) for MP differs from that for OMP in (2.2), we need to establish rotational invariance for $\mathbf{q}_{s\|}^{(\ell)}$ and $\mathbf{q}_{s\perp}^{(\ell)}$, which will be done in Lemma 2.11 below. The bounds on $\mathrm{P}[\mathcal{E}_3^{(\ell,i,s)}]$, $\mathrm{P}[\mathcal{E}_4]$, and $\mathrm{P}[\mathcal{E}_5^{(\ell,i)}]$ in Lemmata 2.2, 2.10, and 2.9, respectively, do not depend on a particular property of $\mathbf{r}_s^{(\ell)}$ apart from $\|\mathbf{r}_s^{(\ell)}\|_2 \leq \|\mathbf{y}_i^{(\ell)}\|_2$ in the case of $\mathrm{P}[\mathcal{E}_3^{(\ell,i,s)}]$ (we have $\|\mathbf{q}_s^{(\ell)}\|_2 \leq \|\mathbf{y}_i^{(\ell)}\|_2$ as a consequence of (Mallat and Zhang, 1993, Eq. 13)). Thus, the bounds on $\mathrm{P}[\mathcal{E}_1^{(\ell,i,s)}]$–$\mathrm{P}[\mathcal{E}_5^{(\ell,i)}]$ continue to hold for $\mathbf{r}_s^{(\ell)}$, $\mathbf{r}_{s\|}^{(\ell)}$, and $\mathbf{r}_{s\perp}^{(\ell)}$ in $\mathcal{E}_1^{(\ell,i,s)}$–$\mathcal{E}_5^{(\ell,i)}$ replaced by $\mathbf{q}_s^{(\ell)}$, $\mathbf{q}_{s\|}^{(\ell)}$, and $\mathbf{q}_{s\perp}^{(\ell)}$, respectively, and we readily get the upper bound (2.18) on the LHS of (2.16) and the lower bound (2.24) on the RHS of (2.16). The bounds on $\|\mathbf{r}_{s\|}^{(\ell)}\|_2$ and $\|\mathbf{r}_{s\perp}^{(\ell)}\|_2$ in $\mathcal{E}_6^{(\ell,i)}$ and $\mathcal{E}_7^{(\ell,i)}$, respectively, in the proof of Theorem 2.1 require more work. In particular, as the stopping behavior of MP is different from that of OMP, we need the corresponding bounds on $\|\mathbf{q}_{s\perp}^{(\ell)}\|_2$ and $\|\mathbf{q}_{s\|}^{(\ell)}\|_2$ to hold for all MP iterations $s \in [s_\mathrm{a}]$ and for a maximum sparsity level of $p_{\max}$. This will be accomplished by deriving an upper bound on $\|\mathbf{q}_{s\perp}^{(\ell)}\|_2$ and a lower bound on $\|\mathbf{q}_{s\|}^{(\ell)}\|_2$ leading to suitably modified events $\tilde{\mathcal{E}}_6^{(\ell,i)}$ and $\tilde{\mathcal{E}}_7^{(\ell,i)}$, respectively, as defined below. Specifically, the resulting upper bound on $\|\mathbf{q}_{s\perp}^{(\ell)}\|_2$ is slightly weaker than that on $\|\mathbf{r}_{s\perp}^{(\ell)}\|_2$ in $\mathcal{E}_6^{(\ell,i)}$, but exhibits the same scaling behavior in σ, whereas the resulting lower bound on $\|\mathbf{q}_{s\|}^{(\ell)}\|_2$ is identical to the one on $\|\mathbf{r}_{s\|}^{(\ell)}\|_2$ in $\mathcal{E}_7^{(\ell,i)}$.

We proceed by introducing the modified events

$$\tilde{\mathcal{E}}_6^{(\ell,i)} := \left\{ \left\|\mathbf{q}_{s\perp}^{(\ell)}\right\|_2 \leq \left\|\mathbf{z}_{i\perp}^{(\ell)}\right\|_2 + \frac{6\sigma}{\tilde{a}}\left\|\mathbf{y}_i^{(\ell)}\right\|_2, \ \forall s \leq s_\mathrm{a} \right\}, \quad \text{and}$$

$$\tilde{\mathcal{E}}_7^{(\ell,i)} := \Bigg\{ \left\|\mathbf{q}_{s\|}^{(\ell)}\right\|_2 \\ > \left\|\mathbf{y}_{i\|}^{(\ell)}\right\|_2 \left(\frac{2}{3} - \sqrt{\frac{3p_{\max}\log((n_\ell-1)e/p_{\max})}{d_\ell}} \right), \forall s \leq s_\mathrm{a} \Bigg\}, \tag{2.77}$$

where $\tilde{a} := \min_{j\in[n_\ell]\setminus\{i\}} \|\mathbf{y}_{j\|}^{(\ell)}\|_2$, and deriving lower bounds on $\mathrm{P}[\tilde{\mathcal{E}}_6^{(\ell,i)}]$ and $\mathrm{P}[\tilde{\mathcal{E}}_7^{(\ell,i)}]$ in Lemmata 2.12 and 2.13, respectively. These lower bounds will turn out to be identical to those for SSC-OMP. Although the corresponding proofs rely on arguments similar in spirit to those used for OMP, the technical details are dissimilar enough to warrant detailed presentation.

We continue by establishing the bounds on $\|\mathbf{q}_{s\|}^{(\ell)}\|_2$ and $\|\mathbf{q}_{s\perp}^{(\ell)}\|_2$, as announced. Using the assumptions $m \geq 2d_{\max}$, $\sigma \leq 1/2$, and $p_{\max} \leq \min_{\ell\in[L]}\{c_s d_\ell / \log((n_\ell - 1)e/p_{\max})\}$, we have on $\mathcal{E}_4 \cap \tilde{\mathcal{E}}_6^{(\ell,i)}$

that

$$\left\|\mathbf{q}_{s\perp}^{(\ell)}\right\|_2 \leq \frac{3}{2}\sigma + 6\sigma\frac{1+\frac{3}{2}\sigma}{1-\frac{3}{2}\frac{\sqrt{d_\ell}}{\sqrt{m}}\sigma} \leq \sigma(15+20\sigma) \tag{2.78}$$

and, by repeating the steps in (2.29), we get $\|\mathbf{q}_{s\|}^{(\ell)}\|_2 > 1/20$ on $\tilde{\mathcal{E}}_7^{(\ell,i)}$. It therefore follows that (2.16) for MP is implied by (2.8) with $c(\sigma) = 17+23\sigma$ on $\tilde{\mathcal{E}}^\star := \bigcap_{\ell,i,s}(\mathcal{E}_1^{(\ell,i,s)} \cap \mathcal{E}_2^{(\ell,i,s)} \cap \mathcal{E}_3^{(\ell,i,s)} \cap \mathcal{E}_4 \cap \mathcal{E}_5^{(\ell,i)} \cap \tilde{\mathcal{E}}_6^{(\ell,i)} \cap \tilde{\mathcal{E}}_7^{(\ell,i)})$. The proof is completed by lower-bounding $\mathrm{P}[\tilde{\mathcal{E}}^\star]$ via a union bound.

We proceed by establishing the rotational invariance properties of $\mathbf{q}_{s\|}^{(\ell)}$ and $\mathbf{q}_{s\perp}^{(\ell)}$ needed to establish the lower bounds on $\mathrm{P}[\tilde{\mathcal{E}}_6^{(\ell,i)}]$ and $\mathrm{P}[\tilde{\mathcal{E}}_7^{(\ell,i)}]$ in Lemmata 2.12 and 2.13, respectively.

Lemma 2.11. *The distributions of $\mathbf{q}_{s\|}^{(\ell)}$ and $\mathbf{q}_{s\perp}^{(\ell)}$ are rotationally invariant on $\mathcal{S}_\ell$ and $\mathcal{S}_\ell^\perp$, respectively, i.e., for unitary transformations $\mathbf{V}^\|, \mathbf{V}^\perp \in \mathbb{R}^{m\times m}$ of the form specified in Lemma 2.1, we have $\mathbf{V}^\|\mathbf{q}_{s\|}^{(\ell)} \sim \mathbf{q}_{s\|}^{(\ell)}$ and $\mathbf{V}^\perp\mathbf{q}_{s\perp}^{(\ell)} \sim \mathbf{q}_{s\perp}^{(\ell)}$.*

Proof. The arguments employed in this proof are similar to those in the proof of the corresponding result for OMP, Lemma 2.1, but the structure of the proof differs as the MP residual $\mathbf{q}_s^{(\ell)}$ can only be expressed recursively, i.e., as a function of previous residuals. In contrast, the reduced OMP residual $\mathbf{r}_s^{(\ell)}$ can be written as the projection of $\mathbf{y}_i^{(\ell)}$ onto the orthogonal complement of the span of the data points indexed by Λ_s. Throughout the proof, $\omega_s(\mathbf{x}, \mathbf{D})$ denotes the index obtained by the MP algorithm in iteration s when applied to $\mathbf{x}$ with the columns of $\mathbf{D}$ as dictionary elements, and $\mathbf{q}_s(\mathbf{x}, \mathbf{D})$ is the corresponding residual.

We first establish results analogous to (2.32) and (2.33). Again, the proof is effected through induction. We start with the inductive step and assume that

$$\mathbf{q}_{s'}(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)}) = \mathbf{V}\mathbf{q}_{s'}(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)}) \tag{2.79}$$

for fixed $s' < s_\mathrm{a}$, for all unitary matrices $\mathbf{V} \in \mathbb{R}^{m\times m}$. For iteration

$s' + 1$ we then have

$$\begin{aligned}\omega_{s'+1}(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)}) &= \underset{j\in[n_\ell]\setminus\{i\}}{\arg\max} \left|\left\langle \mathbf{V}\mathbf{y}_j^{(\ell)}, \mathbf{q}_{s'}(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)})\right\rangle\right| \\ &= \underset{j\in[n_\ell]\setminus\{i\}}{\arg\max} \left|\left\langle \mathbf{y}_j^{(\ell)}, \mathbf{q}_{s'}(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)})\right\rangle\right| \\ &= \omega_{s'+1}(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)}). \end{aligned} \tag{2.80}$$

Now, using the shorthands $\omega_{s'+1}(\mathbf{V})$ and $\omega_{s'+1}$ for $\omega_{s'+1}(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)})$ and $\omega_{s'+1}(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)})$, respectively, we get

$$\begin{aligned}&\mathbf{q}_{s'+1}(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)}) \\ &\quad= \mathbf{q}_{s'}(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)}) \\ &\qquad\quad - \left\langle \mathbf{q}_{s'}(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)}), \frac{\mathbf{V}\mathbf{y}_{\omega_{s'+1}(\mathbf{V})}^{(\ell)}}{\left\|\mathbf{V}\mathbf{y}_{\omega_{s'+1}(\mathbf{V})}^{(\ell)}\right\|_2}\right\rangle \frac{\mathbf{V}\mathbf{y}_{\omega_{s'+1}(\mathbf{V})}^{(\ell)}}{\left\|\mathbf{V}\mathbf{y}_{\omega_{s'+1}(\mathbf{V})}^{(\ell)}\right\|_2} \\ &\quad= \mathbf{q}_{s'}(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)}) - \left\langle \mathbf{q}_{s'}(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)}), \frac{\mathbf{V}\mathbf{y}_{\omega_{s'+1}}^{(\ell)}}{\left\|\mathbf{y}_{\omega_{s'+1}}^{(\ell)}\right\|_2}\right\rangle \frac{\mathbf{V}\mathbf{y}_{\omega_{s'+1}}^{(\ell)}}{\left\|\mathbf{y}_{\omega_{s'+1}}^{(\ell)}\right\|_2} \\ &\quad= \mathbf{V}\left(\mathbf{q}_{s'}(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)}) - \left\langle \mathbf{q}_{s'}(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)}), \frac{\mathbf{y}_{\omega_{s'+1}}^{(\ell)}}{\left\|\mathbf{y}_{\omega_{s'+1}}^{(\ell)}\right\|_2}\right\rangle \frac{\mathbf{y}_{\omega_{s'+1}}^{(\ell)}}{\left\|\mathbf{y}_{\omega_{s'+1}}^{(\ell)}\right\|_2}\right) \\ &\quad= \mathbf{V}\mathbf{q}_{s'+1}(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)}), \end{aligned} \tag{2.81}$$

where the second and third equality follow from (2.80) and (2.79), respectively. Now, the base case is $\mathbf{q}_0(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)}) = \mathbf{V}\mathbf{y}_i^{(\ell)} = \mathbf{V}\mathbf{q}_0(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)})$, and we therefore established that

$$\mathbf{q}_s(\mathbf{V}\mathbf{y}_i^{(\ell)}, \mathbf{V}\mathbf{Y}_{-i}^{(\ell)}) = \mathbf{V}\mathbf{q}_s(\mathbf{y}_i^{(\ell)}, \mathbf{Y}_{-i}^{(\ell)}), \tag{2.82}$$

for all $s \leq s_\mathrm{a}$ and all unitary $\mathbf{V} \in \mathbb{R}^{m\times m}$.

Finally, repeating the steps leading from (2.34) to (2.35) for $\mathbf{q}_s$ and $\mathbf{q}_{s\|}$ instead of $\mathbf{r}_s$ and $\mathbf{r}_{s\|}$, respectively, yields the desired result. □

We continue with the lower bound on $\mathrm{P}[\tilde{\mathcal{E}}_7^{(\ell,i)}]$.

Lemma 2.12. *Let $n_\ell > 1$. We have*

$$\begin{aligned}\mathrm{P}\left[\tilde{\mathcal{E}}_7^{(\ell,i)}\right] = \mathrm{P}\Bigg[&\left\|\mathbf{q}_{s\|}^{(\ell)}\right\|_2 \\ &> \left\|\mathbf{y}_{i\|}^{(\ell)}\right\|_2 \left(\frac{2}{3} - \sqrt{\frac{3p_{\max}\log((n_\ell-1)e/p_{\max})}{d_\ell}}\right), \forall s \le s_{\mathrm{a}}\Bigg] \\ &\ge 1 - e^{-d_\ell/18}. \qquad (2.83)\end{aligned}$$

Proof. We start by recalling that the reduced MP algorithm decomposes $\mathbf{y}_i^{(\ell)}$ according to (see, e.g., (Mallat and Zhang, 1993))

$$\mathbf{y}_i^{(\ell)} = \sum_{s'=1}^{s} \left\langle \mathbf{q}_{s'-1}^{(\ell)}, \frac{\mathbf{y}_{\omega_{s'}}^{(\ell)}}{\left\|\mathbf{y}_{\omega_{s'}}^{(\ell)}\right\|_2} \right\rangle \frac{\mathbf{y}_{\omega_{s'}}^{(\ell)}}{\left\|\mathbf{y}_{\omega_{s'}}^{(\ell)}\right\|_2} + \mathbf{q}_s^{(\ell)}. \qquad (2.84)$$

Denote by Ω_s the set containing the indices of the points in $\mathcal{Y}_\ell \backslash \{\mathbf{y}_i^{(\ell)}\}$ selected during the first s iterations, i.e.,

$$\Omega_s := \{\omega_{s'} \colon s' \in [s]\}. \qquad (2.85)$$

Note that Ω_s may contain fewer than s indices as reduced MP may select one or more points (from $\mathcal{Y}_\ell \backslash \{\mathbf{y}_i^{(\ell)}\}$) repeatedly. With $\mathbf{P}_{\Omega_s}^{\|} := \mathbf{Y}_{\Omega_s\|}^{(\ell)} {\mathbf{Y}_{\Omega_s\|}^{(\ell)}}^{\dagger}$, using (2.84), we get

$$\begin{aligned}\left\|\mathbf{P}_{\|}\mathbf{q}_s^{(\ell)}\right\|_2 &\ge \left\|(\mathbf{I} - \mathbf{P}_{\Omega_s}^{\|})\mathbf{P}_{\|}\mathbf{q}_s^{(\ell)}\right\|_2 \\ &= \left\|(\mathbf{I} - \mathbf{P}_{\Omega_s}^{\|})\mathbf{P}_{\|}\left(\mathbf{y}_i^{(\ell)} \sum_{s'=1}^{s} \left\langle \mathbf{q}_{s'-1}^{(\ell)}, \frac{\mathbf{y}_{\omega_{s'}}^{(\ell)}}{\left\|\mathbf{y}_{\omega_{s'}}^{(\ell)}\right\|_2} \right\rangle \frac{\mathbf{y}_{\omega_{s'}}^{(\ell)}}{\left\|\mathbf{y}_{\omega_{s'}}^{(\ell)}\right\|_2}\right)\right\|_2 \\ &= \Bigg\|(\mathbf{I} - \mathbf{P}_{\Omega_s}^{\|})\mathbf{y}_{i\|}^{(\ell)}\end{aligned}$$

$$- (\mathbf{I} - \mathbf{P}^{\|}_{\Omega_s}) \underbrace{\left(\sum_{s'=1}^{s} \left\langle \mathbf{q}^{(\ell)}_{s'-1}, \frac{\mathbf{y}^{(\ell)}_{\omega_{s'}}}{\left\| \mathbf{y}^{(\ell)}_{\omega_{s'}} \right\|_2} \right\rangle \frac{\mathbf{y}^{(\ell)}_{\omega_{s'}\|}}{\left\| \mathbf{y}^{(\ell)}_{\omega_{s'}} \right\|_2} \right)}_{\in \mathcal{R}(\mathbf{P}^{\|}_{\Omega_s})} \Bigg\|_2$$

$$= \left\| (\mathbf{I} - \mathbf{P}^{\|}_{\Omega_s}) \mathbf{y}^{(\ell)}_{i\|} \right\|_2$$
$$\geq \| (\mathbf{I} - \mathbf{P}^{\|}_{\Omega_{s_\mathrm{a}}}) \mathbf{y}^{(\ell)}_{i\|} \|_2, \tag{2.86}$$

where the last inequality is by $\mathcal{R}(\mathbf{I} - \mathbf{P}^{\|}_{\Omega_{s_\mathrm{a}}}) \subseteq \mathcal{R}(\mathbf{I} - \mathbf{P}^{\|}_{\Omega_s})$, for $s \leq s_\mathrm{a}$. The proof is now completed by replacing Ω_{s_a} in (2.86) by a fixed $\Gamma \in \mathcal{I}$ (recall the definition of $\mathcal{I}$ from (2.40), with $\bar{s}$ in (2.40) replaced by $p_{\max}$) and by lower-bounding $\|(\mathbf{I} - \mathbf{P}^{\|}_{\Gamma})\mathbf{y}^{(\ell)}_{i\|}\|_2$ for all $\Gamma \in \mathcal{I}$ as in the proof of Lemma 2.5 (following the steps starting from (2.45)). □

Next, we lower-bound $\mathrm{P}[\tilde{\mathcal{E}}^{(\ell,i)}_6]$.

Lemma 2.13. *Set* $\tilde{a} \coloneqq \min_{j \in [n_\ell] \setminus \{i\}} \|\mathbf{y}^{(\ell)}_{j\|}\|_2$ *and assume that* $p_{\max} \leq c_1 d_\ell / \log(e(n_\ell - 1)/p_{\max})$ *for a numerical constant* $c_1 > 0$. *Then, we have*

$$\mathrm{P}\left[\tilde{\mathcal{E}}^{(\ell,i)}_6\right] = \mathrm{P}\left[\left\|\mathbf{q}^{(\ell)}_{s\perp}\right\|_2 \leq \left\|\mathbf{z}^{(\ell)}_{i\perp}\right\|_2 + \frac{6\sigma}{\tilde{a}}\left\|\mathbf{y}^{(\ell)}_i\right\|_2, \forall s \leq s_\mathrm{a}\right]$$
$$\geq 1 - 2e^{-c_2 m} - 2e^{-c_3 d_\ell}, \tag{2.87}$$

where $c_2, c_3 > 0$ *are numerical constants.*

Proof. We start by rewriting (2.84) as

$$\mathbf{y}^{(\ell)}_i = \mathbf{Y}^{(\ell)}_{\Omega_s} \mathbf{b}_{\Omega_s} + \mathbf{q}^{(\ell)}_s, \tag{2.88}$$

where Ω_s was defined in (2.85) and $\mathbf{b}_{\Omega_s}$ contains the coefficients of the representation of $\mathbf{y}^{(\ell)}_i$ computed by MP according to (2.4), i.e., $[\mathbf{b}]_\omega = \sum_{s' \colon \omega_{s'} = \omega} \langle \mathbf{y}^{(\ell)}_{\omega_{s'}}, \mathbf{q}^{(\ell)}_{s'-1} \rangle / \|\mathbf{y}^{(\ell)}_{\omega_{s'}}\|_2^2$, $\omega \in \Omega_s$ (this is a direct consequence of (2.4); we do not reflect dependence of $\mathbf{b}$ on i, ℓ for expositional ease). Next, note that

$$\sigma_{\min}(\tilde{\mathbf{A}}^{(\ell)}_{\Omega_s}) \|\mathbf{b}_{\Omega_s}\|_2 \leq \sigma_{\min}(\mathbf{Y}^{(\ell)}_{\Omega_s}) \|\mathbf{b}_{\Omega_s}\|_2$$

$$\begin{aligned} &\leq \left\| \mathbf{Y}_{\Omega_s}^{(\ell)} \mathbf{b}_{\Omega_s} \right\|_2 = \left\| \mathbf{y}_i^{(\ell)} - \mathbf{q}_s^{(\ell)} \right\|_2 \\ &\leq \left\| \mathbf{y}_i^{(\ell)} \right\|_2 + \left\| \mathbf{q}_s^{(\ell)} \right\|_2 \leq 2 \left\| \mathbf{y}_i^{(\ell)} \right\|_2, \end{aligned} \tag{2.89}$$

where the first inequality is a consequence of $\|\mathbf{Y}_{\Omega_s}^{(\ell)}\mathbf{v}\|_2 \geq \|\mathbf{P}_{\|}\mathbf{Y}_{\Omega_s}^{(\ell)}\mathbf{v}\|_2 = \|\mathbf{P}_{\|}\mathbf{U}^{(\ell)}\tilde{\mathbf{A}}_{\Omega_s}^{(\ell)}\mathbf{v}\|_2 = \|\tilde{\mathbf{A}}_{\Omega_s}^{(\ell)}\mathbf{v}\|_2$, for all $\mathbf{v} \in \mathbb{R}^{|\Omega_s|}$, and the last inequality follows from $\|\mathbf{q}_s^{(\ell)}\|_2 \leq \|\mathbf{y}_i^{(\ell)}\|_2$ (Mallat and Zhang, 1993, Eq. 13). For $\sigma_{\min}(\tilde{\mathbf{A}}_{\Omega_s}^{(\ell)}) > 0$ (we will justify below that $\sigma_{\min}(\tilde{\mathbf{A}}_{\Omega_s}^{(\ell)})$ is, indeed, bounded away from 0 with high probability) it hence follows that $\|\mathbf{b}_{\Omega_s}\|_2 \leq (2/\sigma_{\min}(\tilde{\mathbf{A}}_{\Omega_s}^{(\ell)}))\|\mathbf{y}_i^{(\ell)}\|_2$ and therefore, together with (2.88), we get

$$\begin{aligned} \left\| \mathbf{q}_{s\perp}^{(\ell)} \right\|_2 &= \left\| \mathbf{y}_{i\perp}^{(\ell)} - \mathbf{Y}_{\Omega_s\perp}^{(\ell)} \mathbf{b}_{\Omega_s} \right\|_2 \\ &\leq \left\| \mathbf{y}_{i\perp}^{(\ell)} \right\|_2 + \left\| \mathbf{Y}_{\Omega_s\perp}^{(\ell)} \mathbf{b}_{\Omega_s} \right\|_2 \\ &\leq \left\| \mathbf{y}_{i\perp}^{(\ell)} \right\|_2 + \left\| \mathbf{Y}_{\Omega_s\perp}^{(\ell)} \right\|_{2\to 2} \|\mathbf{b}_{\Omega_s}\|_2 \\ &= \left\| \mathbf{z}_{i\perp}^{(\ell)} \right\|_2 + \left\| \mathbf{Z}_{\Omega_s\perp}^{(\ell)} \right\|_{2\to 2} \|\mathbf{b}_{\Omega_s}\|_2 \\ &\leq \left\| \mathbf{z}_{i\perp}^{(\ell)} \right\|_2 + \frac{2}{\sigma_{\min}(\tilde{\mathbf{A}}_{\Omega_s}^{(\ell)})} \left\| \mathbf{Z}_{\Omega_s\perp}^{(\ell)} \right\|_{2\to 2} \left\| \mathbf{y}_i^{(\ell)} \right\|_2. \end{aligned} \tag{2.90}$$

Replacing Ω_s in (2.90) by a fixed $\Gamma \in \mathcal{J}$ (recall the definition of $\mathcal{J}$ from (2.57), with $s_{\max}$ in (2.57) replaced by $p_{\max}$), (2.90) and (2.60) are equal up to the factor 2 in the second term of (2.90). This factor-of-two difference stems from the update rule for $\mathbf{q}_s^{(\ell)}$ differing from that for $\mathbf{r}_s^{(\ell)}$ and leads to the difference in $c(\sigma)$ between Theorems 2.2 and 2.1. We proceed as in the proof of Lemma 2.13 (starting from (2.58)–(2.60)) by establishing bounds on the tail probabilities of $\|\mathbf{Z}_{\Gamma\perp}^{(\ell)}\|_{2\to 2}$ and $\sigma_{\min}(\tilde{\mathbf{A}}_{\Gamma}^{(\ell)})$, for all $\Gamma \in \mathcal{J}$, via (2.63) and (2.64), respectively, to obtain the result in Lemma 2.13. □

2.C. PROOF OF THEOREM 2.3

We prove the result for OMP. The proof for MP follows simply by replacing the lower bound on $\|\mathbf{r}_{s\|}^{(\ell)}\|_2$ in $\mathcal{E}_7^{(\ell,i)}$ by the lower bound on $\|\mathbf{q}_{s\|}^{(\ell)}\|_2$ in $\tilde{\mathcal{E}}_7^{(\ell,i)}$ (defined in (2.77)), and by noting that $\mathrm{P}[\mathcal{E}_7^{(\ell,i)}] = \mathrm{P}[\tilde{\mathcal{E}}_7^{(\ell,i)}]$.

We only need to address the case

$$\frac{d_\ell}{\log((n_\ell - 1)e)} \min\left\{\frac{1}{3}\left(\frac{2}{3} - \frac{\tau}{1 - \frac{3}{2}\frac{\sqrt{d_\ell}}{\sqrt{m}}\sigma}\right)^2, c_s\right\} \geq 1, \tag{2.91}$$

as otherwise there is nothing to prove. We start by noting that under the conditions of Theorem 2.3 (which are identical to the conditions of Theorem 2.1 minus the condition $s_{\max} \leq \min_{\ell \in [L]}\{c_s d_\ell / \log((n_\ell - 1)e/s_{\max})\}$), it follows from the proof of Theorem 2.1 that, conditionally on $\mathcal{E}^\star$ as defined in (2.30), reduced OMP and OMP are guaranteed to select the same points to represent $\mathbf{y}_i^{(\ell)}$ during the first $\lfloor c_s d_\ell / \log((n_\ell - 1)e) \rfloor \geq$ (2.10) iterations, for all $i \in [n_\ell]$, $\ell \in [L]$. Therefore, conditionally on $\mathcal{E}^\star$, for τ small enough reduced OMP will perform a number of iterations lower-bounded by (2.10), for all $\mathbf{y}_i^{(\ell)}$, $i \in [n_\ell]$, $\ell \in [L]$, which implies that the number of points from $\mathcal{Y}_\ell \backslash \{\mathbf{y}_i^{(\ell)}\}$ selected by OMP is lower-bounded by (2.10) as well, for all $\mathbf{y}_i^{(\ell)}$, $i \in [n_\ell]$, $\ell \in [L]$. Specifically, we will show that for $\tau \in [0, 2/3 - (\sqrt{d_{\max}}/\sqrt{m})\sigma]$ the number of reduced OMP iterations is lower-bounded by (2.10). This will be accomplished by establishing that for $\tau \in [0, 2/3 - (\sqrt{d_{\max}}/\sqrt{m})\sigma]$, on $\mathcal{E}^\star$, we have $\|\mathbf{r}_{s_\tau}^{(\ell)}\|_2 > \tau$ for all $\mathbf{y}_i^{(\ell)}$, $i \in [n_\ell]$, $\ell \in [L]$, where

$$s_\tau := \left\lfloor \frac{d_\ell}{\log((n_\ell - 1)e)} \frac{1}{3}\left(\frac{2}{3} - \frac{\tau}{1 - \frac{3}{2}\frac{\sqrt{d_\ell}}{\sqrt{m}}\sigma}\right)^2 \right\rfloor. \tag{2.92}$$

Indeed, as the stopping criterion $\max_{\mathbf{y}_j^{(\ell)} \in \mathcal{Y}_\ell \backslash (\Lambda_s \cup \{\mathbf{y}_i^{(\ell)}\})} |\langle \mathbf{y}_j^{(\ell)}, \mathbf{r}_s^{(\ell)} \rangle| =$

0 is activated only after $\min\{m, n_\ell - 1\} > d_\ell > s_\tau$ iterations (as the points in $\mathcal{Y}_\ell$ are in general position w.p. 1), $\|\mathbf{r}_{s_\tau}^{(\ell)}\|_2 > \tau$ implies that reduced OMP performs at least s_τ iterations. On the event $\mathcal{E}_4 \cap \mathcal{E}_7^{(\ell,i)} \supset \mathcal{E}^\star$ ($\mathcal{E}_4$ and $\mathcal{E}_7^{(\ell,i)}$ are defined in (2.22) and (2.27), respectively), setting $\bar{s} = s_\tau$ in $\mathcal{E}_7^{(\ell,i)}$, we have

$$\begin{aligned}
\left\|\mathbf{r}_{\bar{s}}^{(\ell)}\right\|_2 &\geq \left\|\mathbf{r}_{\bar{s}\|}^{(\ell)}\right\|_2 \\
&> \left\|\mathbf{y}_{i\|}^{(\ell)}\right\|_2 \left(\frac{2}{3} - \sqrt{\frac{3\bar{s}\log((n_\ell - 1)e/\bar{s})}{d_\ell}}\right) \\
&\geq \left(1 - \frac{3}{2}\frac{\sqrt{d_\ell}}{\sqrt{m}}\sigma\right)\left(\frac{2}{3} - \sqrt{\frac{3\bar{s}\log((n_\ell - 1)e/\bar{s})}{d_\ell}}\right) \\
&\geq \left(1 - \frac{3}{2}\frac{\sqrt{d_\ell}}{\sqrt{m}}\sigma\right)\left(\frac{2}{3}\right. \\
&\qquad \left. - \left(\left\lfloor \frac{d_\ell}{\log((n_\ell - 1)e)}\frac{1}{3}\left(\frac{2}{3} - \frac{\tau}{1 - \frac{3}{2}\frac{\sqrt{d_\ell}}{\sqrt{m}}\sigma}\right)^2\right\rfloor \right.\right. \\
&\qquad\qquad \left.\left. \cdot \frac{3\log((n_\ell - 1)e)}{d_\ell}\right)^{\frac{1}{2}}\right) \\
&\geq \left(1 - \frac{3}{2}\frac{\sqrt{d_\ell}}{\sqrt{m}}\sigma\right)\left(\frac{2}{3} - \left(\frac{2}{3} - \frac{\tau}{1 - \frac{3}{2}\frac{\sqrt{d_\ell}}{\sqrt{m}}\sigma}\right)\right) \\
&= \tau,
\end{aligned} \tag{2.93}$$

where for the second inequality we used that $\|\mathbf{y}_{i\|}^{(\ell)}\|_2 \geq \|\mathbf{x}_i^{(\ell)}\|_2 - \|\mathbf{z}_{i\|}^{(\ell)}\|_2 \geq 1 - 3\sqrt{d_\ell}/(2\sqrt{m})$ on $\mathcal{E}_4$, $\log((n_\ell - 1)e/\bar{s}) \leq \log((n_\ell - 1)e)$, for $\bar{s} \geq 1$, for the third inequality, and $\tau \leq 2/3 - (\sqrt{d_{\max}}/\sqrt{m})\sigma$ for the last inequality. This completes the proof.

2.D. SUPPLEMENTARY NOTES

Lemma 2.14 (Vershynin (2012), Lem. 5.24)**.** *Let $X_1, \dots, X_n$ be independent centered sub-gaussian random variables (see footnote 4). Then, $X = (X_1, \dots, X_n)$ is a centered sub-gaussian random vector in $\mathbb{R}^n$, and*

$$\|X\|_{\Psi_2} \le C \max_{i \in [n]} \|X_i\|_{\Psi_2}, \tag{2.94}$$

where C is a numerical constant.

Theorem 2.5 (Vershynin (2012), Lem. 5.39)**.** *Let $\mathbf{A}$ be an $m \times n$ matrix whose rows are independent sub-gaussian isotropic random vectors (Vershynin, 2012, Def. 5.19, Def. 5.22) in $\mathbb{R}^n$. Then, for $t \ge 0$, w.p. at least $1 - 2e^{-ct^2}$, we have*

$$\sqrt{m} - C\sqrt{n} - t \le \sigma_{\min}(\mathbf{A}) \le \sigma_{\max}(\mathbf{A}) \le \sqrt{m} + C\sqrt{n} + t. \tag{2.95}$$

Here $C = C_K, c = c_K > 0$ depend only on the sub-gaussian norm $K = \max_i \|\mathbf{A}_{:,i}\|_{\Psi_2}$ of the rows of $\mathbf{A}$.

Theorem 2.6 (Vershynin (2012), Cor. 5.35)**.** *Let $\mathbf{A}$ be an $m \times n$ matrix with i.i.d. standard normal entries. Then, for $t \ge 0$, w.p. at least $1 - 2e^{-t^2/2}$, we have*

$$\sqrt{m} - \sqrt{n} - t \le \sigma_{\min}(\mathbf{A}) \le \sigma_{\max}(\mathbf{A}) \le \sqrt{m} + \sqrt{n} + t. \tag{2.96}$$

CHAPTER 3

Robust Nonparametric Nearest Neighbor Random Process Clustering

We consider the problem of clustering noisy finite-length observations of stationary ergodic random processes according to their generative models without prior knowledge of the model statistics and the number of generative models. Two algorithms, both using the L^1-distance between estimated PSDs as a measure of dissimilarity, are analyzed. The first one, termed NNPC, relies on partitioning the nearest neighbor graph of the observations via spectral clustering. The second algorithm, simply referred to as KM, consists of a single k-means iteration with farthest point initialization and was considered before in the literature, albeit with a different dissimilarity measure and with asymptotic performance results only. We prove that both algorithms succeed with high probability in the presence of noise and missing entries, and even when the generative process PSDs overlap significantly, all provided that the observation length is sufficiently large. Our results quantify the tradeoff between the overlap of the generative process PSDs, the observation length, the fraction of missing entries, and the noise variance. Furthermore, we prove that treating the finite-length observations of stationary ergodic random processes as vectors in Euclidean space and clustering them using the TSC algorithm, the subspace clustering cousin of NNPC, results in performance strictly inferior to that of NNPC. We argue that the underlying cause is to be found in TSC employing spherical distance as dissimilarity measure, thereby ignoring the stationary process structure of the observations. Finally, we provide extensive numerical results for synthetic and real data and find that NNPC outperforms state-of-the-art algorithms in human motion sequence clustering.

3.1. FORMAL PROBLEM STATEMENT AND ALGORITHMS

We consider the following clustering problem. Given the unlabeled data set $\mathcal{X} = \mathcal{X}_1 \cup \ldots \cup \mathcal{X}_L$ of cardinality N, where $\mathcal{X}_\ell = \{x_i^{(\ell)}\}_{i=1}^{n_\ell}$ contains noisy length-M observations $x_i^{(\ell)}$—possibly with missing entries—of the real-valued discrete-time stationary ergodic random process $X^{(\ell)}[m]$, $m \in \mathbb{Z}$, corresponding to the ℓth generative model, find the partition $\mathcal{X}_1, \ldots, \mathcal{X}_L$. The statistics of the generative models and of the noise processes, and the number of generative models, are all assumed unknown.

Both clustering algorithms considered in this chapter are based on the following measure for the distance between pairs of processes. With the PSD of $X^{(\ell)}$ denoted by $s^{(\ell)}(f)$, $f \in [0, 1)$, we define the distance (dissimilarity) between the processes $X^{(k)}$ and $X^{(\ell)}$ as $d(X^{(k)}, X^{(\ell)}) := \frac{1}{2} \int_0^1 |s^{(k)}(f) - s^{(\ell)}(f)| \mathrm{d}f$. As argued below, for the algorithms to be meaningful, the different processes have to be of the same or at least of comparable power, which motivates the normalization $\int_0^1 s^{(\ell)}(f) \mathrm{d}f = 1$, $\ell \in [L]$. Now, this implies that $d(X^{(k)}, X^{(\ell)}) \leq \frac{1}{2} \int_0^1 |s^{(k)}(f)| \mathrm{d}f + \frac{1}{2} \int_0^1 |s^{(\ell)}(f)| \mathrm{d}f = \frac{1}{2} \int_0^1 s^{(k)}(f) \mathrm{d}f + \frac{1}{2} \int_0^1 s^{(\ell)}(f) \mathrm{d}f = 1$, and hence $d(X^{(k)}, X^{(\ell)}) \in [0, 1]$. The distance measure $d(X^{(k)}, X^{(\ell)})$ is close to 1 when $s^{(k)}$ and $s^{(\ell)}$ are concentrated on disjoint frequency bands and close to 0 when they exhibit similar support sets and shapes. In contrast, for general L^p-distances $d_{L^p}(X^{(k)}, X^{(\ell)}) := (\int_0^1 |s^{(k)}(f) - s^{(\ell)}(f)|^p \mathrm{d}f)^{\frac{1}{p}}$, with $p > 1$, it is easy to see that $\int_0^1 s^{(\ell)}(f) \mathrm{d}f = 1$, $\ell \in [L]$, does not imply a uniform upper bound for $d_{L^p}(X^{(k)}, X^{(\ell)})$. For example, $d_{L^\infty}(X^{(k)}, X^{(\ell)})$ can become arbitrarily large if we set

$s^{(k)}(f) = 1$, $f \in [0,1)$, and let $s^{(\ell)}$ have a sharp peak at some frequency $f_0 \in [0,1)$, while maintaining $\int_0^1 s^{(\ell)}(f)\mathrm{d}f = 1$.

We now present the NNPC and the KM algorithms. Recall that NNPC is inspired by the TSC algorithm introduced in (Heckel and Bölcskei, 2015), and KM is obtained by replacing the distance measure in Algorithm 1 in (Ryabko, 2010) by the distance measure d defined above. In principle, NNPC and KM are applicable to general (real-valued) time series, in particular also to non-stationary random processes, but the definition of d above is obviously motivated by stationarity.

The NNPC algorithm: *Given a set $\mathcal{X}$ of N length-M observations, the number of generative models, L (the estimation of L from $\mathcal{X}$ is discussed below), and the parameter q, carry out the following steps.*

Step 1: *For every $x_i \in \mathcal{X}$, estimate the PSD $\hat{s}_i(f)$ via the Blackman-Tukey (BT) estimator according to*

$$\hat{s}_i(f) := \sum_{m=-M+1}^{M-1} g[m]\hat{r}_i[m]e^{-\mathrm{i}2\pi fm}, \quad \textit{where} \tag{3.1}$$

$$\hat{r}_i[m] := \frac{1}{M}\sum_{n=0}^{M-|m|-1} x_i[n+m]x_i[n], \quad |m| \le M-1,$$

and $g[m]$, $m \in \mathbb{Z}$, is an even window function (i.e., $g[m] = g[-m]$) with $g[m] = 0$ for $|m| \ge M$, and with bounded non-negative discrete-time Fourier transform (DTFT).

Step 2: *For every $x_i \in \mathcal{X}$, identify the set $\mathcal{T}_i \subset [N]\backslash\{i\}$ of cardinality q defined through*

$$d(x_i, x_j) \le d(x_i, x_v), \quad \textit{for all}\, j \in \mathcal{T}_i \textit{ and all}\, v \notin \mathcal{T}_i,$$

where

$$d(x_i, x_j) := \frac{1}{2}\int_0^1 |\hat{s}_i(f) - \hat{s}_j(f)|\,\mathrm{d}f. \tag{3.2}$$

Step 3: *Let* $\mathbf{z}_j \in \mathbb{R}^N$ *be the vector with* i*th entry* $\exp(-2\,d(x_i, x_j))$, *if* $i \in \mathcal{T}_j$, *and* 0, *if* $i \notin \mathcal{T}_j$.

Step 4: *Construct the adjacency matrix* $\mathbf{A}$ *according to* $\mathbf{A} = \mathbf{Z} + \mathbf{Z}^\top$, *where* $\mathbf{Z} = [\mathbf{z}_1 \; \ldots \; \mathbf{z}_N]$.

Step 5: *Apply normalized spectral clustering (von Luxburg, 2007) to* $(\mathbf{A}, L)$.

Step 2 of NNPC determines the q-nearest neighbors of every observation w.r.t. to the distance measure d. We henceforth denote the corresponding nearest neighbor graph with adjacency matrix $\mathbf{A}$ constructed in Step 4 by G. The parameter q determines the minimum degree of G. Choosing q too small results in the observations stemming from a given generative model forming multiple connected components in G and hence not being assigned to the same cluster in Step 5. This problem can be countered by taking q larger, which, however, increases the chances of observations originating from different generative models being connected in G, thereby increasing the likelihood of incorrect cluster assignments. These tradeoffs are identical to those associated with the choice of the parameter q in TSC (Heckel and Bölcskei, 2015). Note that spectral clustering is robust in the sense that it may deliver correct clustering even when G contains edges connecting observations that originate from different generative models, as long as the corresponding edge weights are sufficiently small.

The number of generative models, L, may be estimated in Step 4 based on the adjacency matrix $\mathbf{A}$ using the *eigengap heuristic* (von Luxburg, 2007) (note that L is needed only in Step 5), which relies on the fact that the number of zero eigenvalues of the normalized Laplacian of G equals the number of connected components in G.

The KM algorithm (Ryabko, 2010)**:** *Given a set* $\mathcal{X}$ *of* N *length-*M *observations and the number of generative models* L, *carry out the following steps.*

Step 1: *Initialize* $c_1 := 1$ *and* $\hat{\mathcal{X}}_\ell := \{\}$, *for all* $\ell \in [L]$.

Step 2: *For every $x_i \in \mathcal{X}$, estimate the PSD $\hat{s}_i(f)$ via the BT estimator* (3.1)*.*

Step 3: for $p = 2$ to L do:

$$c_p := \arg\max_{i \in [N]} \big(\min_{\ell \in [p-1]} d(x_i, x_{c_\ell}) \big),$$

with d as defined in (3.2)*.*

Step 4: for $i = 1$ to N do:

$$\ell^\star \leftarrow \arg\min_{\ell \in [L]} d(x_i, x_{c_\ell}) \tag{3.3}$$

$$\hat{\mathcal{X}}_{\ell^\star} \leftarrow \hat{\mathcal{X}}_{\ell^\star} \cup \{x_i\} \tag{3.4}$$

KM selects the cluster centers in Step 3 and determines the assignments of the observations to these cluster centers in Step 4. Specifically, the algorithm selects x_1 as the first cluster center and then recursively determines the remaining cluster centers by maximizing the minimum distance to the cluster centers already chosen. In Step 4, it then assigns each observation to the closest cluster center (see Figure 3.1). Intuitively, KM recovers the correct cluster assignments if the clusters are separated well enough. In practice, performing additional k-means iterations by alternating between cluster center refinement (simply by taking the refined center to be the average of the observations assigned to it) and re-assignment of the data points to the refined cluster centers, can often improve performance. Numerical results on the effect of additional k-means iterations are provided in Section 3.4. Our analytical results, however, all pertain to the case of a single k-means iteration per the definition of the KM algorithm above. Note that besides the number of clusters, L, KM does not have other parameters such as q in NNPC.

Both NNPC and KM are based on comparisons of distances between observations, and are, therefore, meaningful only if the underlying processes $X^{(\ell)}$ are of comparable power $\int_0^1 s^{(\ell)}(f)\mathrm{d}f$. Indeed, when this

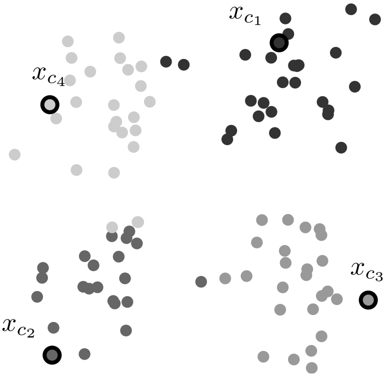


Fig. 3.1: Clustering of an example data set in $\mathbb{R}^2$ determined by KM with farthest point intialization and based on Euclidean distance.

is not the case, the distance between the observations is determined predominantly by the difference in power rather than the difference in PSD support sets and shapes. Note that the assumption of comparable power is not critical as we can normalize the observations in practice.

The choice of the window function g in (3.1) determines the bias-variance tradeoff of the BT estimator and through the distance estimates $d(x_i, x_j)$ ultimately the bias-variance tradeoff of NNPC and KM. For a discussion of window choice considerations for the BT estimator in a general context, we refer the reader to (Stoica and Moses, 2005, Sec. 2.6). We only remark here that the variance of the BT estimator goes to 0 as $M \to \infty$ under rather mild conditions on the process PSD and for $g \in \ell_1$ (Kay, 1988, Appendix B4); the statistical data model employed in this chapter (and described in the next section) satisfies these conditions on the PSDs.

Next, we briefly discuss computational aspects of NNPC and KM for $L \ll N$, the situation typically encountered in practice. The BT PSD estimates (3.1) can be computed efficiently using the FFT. NNPC is a spectral clustering algorithm and as such requires the $N(N-1)/2$ distances between all pairs of observations to construct G. NNPC furthermore needs to determine the L eigenvectors corresponding to the L smallest eigenvalues of the $N \times N$ normalized graph Laplacian, which requires $O(N^3)$ operations (without exploiting potentially present structural properties of the Laplacian such as, e.g., sparsity). Spectral clustering then performs standard k-means clustering on the rows of the resulting $N \times L$ matrix of eigenvectors. The computational complexity of NNPC therefore becomes challenging for large N. Several spectral clustering methods suitable for data sets of up to millions of observations are available in the literature, see, e.g., (Yan et al., 2009; Li et al., 2011; Chen et al., 2011). KM, on the other hand, is computationally considerably less expensive, requiring only $O(NL^2)$ distance computations.

We finally note that both NNPC and KM along with the corresponding analytical performance guarantees presented in the next section can easily be generalized to stationary ergodic vector-processes $\mathbf{x}^{(\ell)}[m] \in \mathbb{R}^n$, $m \in \mathbb{Z}$. Specifically, with the spectral density matrices $\mathbf{S}^{(\ell)}(f) := \sum_{m=-\infty}^{\infty} \mathbb{E}\big[\mathbf{x}^{(\ell)}[m](\mathbf{x}^{(\ell)}[0])^\top\big] e^{-\mathrm{i}2\pi f m} \in \mathbb{R}^{n \times n}$, $\ell \in [L]$, one defines the distance measure

$$d(\mathbf{x}^{(k)}, \mathbf{x}^{(\ell)}) = \sum_{u,v \in [n]} \int_0^1 |\mathbf{S}_{u,v}^{(k)}(f) - \mathbf{S}_{u,v}^{(\ell)}(f)| \, \mathrm{d}f$$

and employs the BT estimator in (3.1) component-wise to estimate $\mathbf{S}^{(\ell)}(f)$. As this requires the computation of distances between all scalar random process components, evaluating $d(\mathbf{x}^{(k)}, \mathbf{x}^{(\ell)})$ in the vector case incurs $n(n+1)/2$ (exploiting the symmetry of $\mathbf{S}^{(\ell)}(f)$) times the cost in the scalar case. All other steps of NNPC and KM remain unchanged and hence have the same computational complexity as in the scalar case. For simplicity of exposition, we focus on the scalar case throughout the chapter.

Relation to prior work: Numerical studies of time series clustering based on spectral clustering of the q-nearest neighbor graph using different dissimilarity measures (albeit not the L^1-distance, or, for that matter, other L^p-distances, between estimated PSDs) were reported in (Tucci and Raugi, 2011). In (Ferreira and Zhao, 2016) time series clustering is formulated as a community detection problem in graphs, but no analytical performance results are provided. KM with distributional distance as dissimilarity measure was proven in (Ryabko, 2010)—for more general (i.e., not necessarily Gaussian) generative models—to deliver correct clustering with probability approaching 1 as the observation length goes to infinity. We note, however, that estimating the distributional distance is computationally more demanding than estimating the L^1-distance between PSDs.

3.2. ANALYTICAL PERFORMANCE RESULTS

We start by describing the statistical data model underlying our analytical performance results. Recall that both NNPC and KM are, in principle, applicable to general real-valued time series including non-stationary processes. The performance analysis conducted here applies, however, to stationary processes. In addition, we take into account additive noise and potentially missing entries. Specifically, we assume that the $x_i^{(\ell)}$ are obtained as contiguous length-M observations of $\check{X}^{(\ell)}[m] := U^{(\ell)}[m](X^{(\ell)}[m] + W^{(\ell)}[m]), m \in \mathbb{Z}$, where $U^{(\ell)}$ is a Bernoulli process with i.i.d. entries according to $\mathrm{P}\big[U^{(\ell)}[m] = 1\big] = 1 - \mathrm{P}\big[U^{(\ell)}[m] = 0\big] = p > 0$ (we henceforth refer to p as sampling probability), $X^{(\ell)}$ is zero-mean stationary Gaussian with PSD $s^{(\ell)}(f)$, and $W^{(\ell)}$ is a zero-mean white Gaussian noise process with variance σ^2. The autocorrelation functions (ACFs) $r^{(\ell)}[m] := \int_0^1 s^{(\ell)}(f)e^{\mathrm{i}2\pi f m}\mathrm{d}f$ of the $X^{(\ell)}$ are assumed absolutely summable, i.e., $\sum_{m=-\infty}^{\infty} |r^{(\ell)}[m]| < \infty$, $\ell \in [L]$, which implies continuity of the $s^{(\ell)}(f)$ and thereby ergodicity of the corresponding processes $X^{(\ell)}$ (Maruyama, 1949). Moreover, we take the PSDs to be normalized according to $\int_0^1 s^{(\ell)}(f)\mathrm{d}f = 1$, $\ell \in [L]$, and we let $B := \max_{\ell \in [L]} \sup_{f \in [0,1)} s^{(\ell)}(f)$. We further assume that $U^{(\ell)}$, $X^{(\ell)}$, and $W^{(\ell)}$ are mutually independent. As a consequence, the noisy process $\tilde{X}^{(\ell)}[m] := X^{(\ell)}[m] + W^{(\ell)}[m]$ and the Bernoulli process $U^{(\ell)}$ are jointly stationary ergodic so that $\check{X}^{(\ell)}[m] = U^{(\ell)}[m]\tilde{X}^{(\ell)}[m]$ is stationary ergodic by (White, 2014, Prop. 3.36). Furthermore, we denote the ACF of the noisy process $\tilde{X}^{(\ell)}[m]$ by $\tilde{r}^{(\ell)}[m]$ and note that $\tilde{r}^{(\ell)}[m] = r^{(\ell)}[m] + \sigma^2\delta[m]$. It follows from $\check{X}^{(\ell)}[m] = U^{(\ell)}[m]\tilde{X}^{(\ell)}[m]$ that $\check{r}^{(\ell)}[m] = u[m]\tilde{r}^{(\ell)}[m]$, where $u[m] := p$ for $m = 0$, and $u[m] := p^2$, else. For each ℓ, the $x_i^{(\ell)}$ may either stem from independent realizations of $\check{X}^{(\ell)}$ or correspond to different (possibly overlapping) length-M segments of a given realization of $\check{X}^{(\ell)}$. In the latter case the $x_i^{(\ell)}$ will not be statistically independent in general. This is, however, not an issue as statistical independence is not required in our analysis, neither across observations stemming from a given generative model nor across observations originating from different generative models.

Multiplication of $\tilde{X}^{(\ell)}$ by the Bernoulli process $U^{(\ell)}$ models, e.g., a sampling device which acquires only every $(1/p)$th sample on average. Moreover, in practice we could deliberately subsample in order to speed up the computation of the distances d when the observation length M is large. Specifically, with observation length M and sampling probability p, we get $\approx (1-p)M$ entries of $x_i^{(\ell)}$ that are set to 0, which can be exploited when computing the BT estimates using the FFT (Skinner, 1976).

Naïvely applying the BT estimator to the $x_i^{(\ell)}$ delivers PSD estimates that, owing to $\check{r}^{(\ell)}[m] = u[m]\tilde{r}^{(\ell)}[m]$, can be severely biased compared to estimates that would be obtained from observations with no missing entries. Indeed, as $\check{r}^{(\ell)}[0] = p\,\tilde{r}^{(\ell)}[0]$ and $\check{r}^{(\ell)}[m] = p^2\tilde{r}^{(\ell)}[m]$, for $m \neq 0$, for small p, $u[m]$ assigns a much larger weight to lag $m = 0$ than to the lags $m \neq 0$. To correct this bias, we assume in the remainder of the chapter (in particular also in the analytical results below) that the BT estimates in (3.1) are computed for the window function $\hat{g}[m] := g[m]/u[m]$, $m \in \mathbb{Z}$, i.e., g in NNPC and KM is replaced by $\hat{g}$. While $\hat{g}$ remains even and supported on $\{-M+1, \dots, M-1\}$, BT PSD estimates based on $\hat{g}$ are not guaranteed to be non-negative (in contrast to estimates based on g directly (Stoica and Moses, 2005, Sec. 2.5.2)) as the DTFT of $\hat{g}$ may not be non-negative. This is, however, not an issue as we consider distances between PSDs only and do not explicitly make use of the positivity property of PSDs. We note that bias correction requires knowledge of p, which can be obtained in practice simply by estimating the average number of non-zero entries in the $x_i^{(\ell)}$. In addition, we will assume that $g[0] = 1$ and g has a bounded DTFT $g(f)$, i.e., $0 \leq g(f) \leq A < \infty$, $f \in [0, 1)$. An example of such a window function is the Bartlett window (see (3.12)) used in the experiments in Section 3.4. Our performance results will be seen to depend on the maximum ACF moment $\mu_{\max} := \max_{\ell \in [L]} \mu^{(\ell)}$, where $\mu^{(\ell)} := \sum_{m=-\infty}^{\infty} |h[m]||r^{(\ell)}[m]|$ with

$$h[m] := \begin{cases} 1 - g[m](1 - |m|/M), & \text{for } |m| < M \\ 1, & \text{otherwise.} \end{cases} \tag{3.5}$$

We are now ready to state our main results. For the NNPC algorithm, we provide a sufficient condition for the NFC property to hold. The notion of NFC property used here is identical to that used in Chapter 2 for subspace clustering, defined in Definition 2.1. For completeness we restate it here for NNPC. Recall that G is the nearest neighbor graph with adjacency matrix $\mathbf{A}$, as constructed in Step 4 of NNPC.

Definition 3.1 (No false connections (NFC) property). *The graph G satisfies the no false connections property if, for all $\ell \in [L]$, all nodes corresponding to $\mathcal{X}_\ell$ are connected exclusively to nodes corresponding to $\mathcal{X}_\ell$.*

We henceforth say that "NNPC succeeds" if the NFC property is satisfied. Recall that, although the NFC property alone does not guarantee correct clustering, it was found to be a sensible performance measure for subspace clustering algorithms (see, e.g., (Elhamifar and Vidal, 2013; Soltanolkotabi et al., 2014; Heckel and Bölcskei, 2015) and further references in Section 2.2). To ensure correct clustering one would additionally need the subgraph of G corresponding to $\mathcal{X}_\ell$ to be connected, for each $\ell \in [L]$ (von Luxburg, 2007, Prop. 4; Sec. 7). Establishing conditions for this to hold appears to be difficult, at least for the statistical data model considered here.

Theorem 3.1. *Let $\mathcal{X}$ be generated according to the statistical data model described above and assume that $q \leq \min_{\ell \in [L]}(n_\ell - 1)$. Then,* *the clustering condition*

$$\min_{\substack{k,\ell\in[L]:\\ k\neq\ell}} d(X^{(k)},X^{(\ell)}) > \frac{8\sqrt{2}A(B+\sigma^2+\sqrt{2}(1+p)(1+\sigma^2))}{p^2}\sqrt{\frac{\log M}{M}}+2\mu_{\max} \qquad (3.6)$$

guarantees that G satisfies the NFC property with probability at least $1-6N/M^2$.

The condition $q \leq \min_{\ell\in[L]}(n_\ell - 1)$ is necessary for the NFC property to hold as choosing $q > \min_{\ell\in[L]}(n_\ell - 1)$ would force NNPC to select observations from $\mathcal{X}\backslash\mathcal{X}_\ell$ for at least one of the data points $x_i^{(\ell)}$. As the n_ℓ are unknown in practice, one has to guess q while taking into account the tradeoffs related to the choice of q as discussed in Section 3.1.

Our main result for KM comes with a performance guarantee that is stronger than the NFC property, namely it ensures correct clustering; accordingly, "KM succeeds" henceforth refers to KM delivering correct clustering. This stronger result is possible as KM does not entail a spectral clustering step and is therefore much easier to analyze. On the other hand, NNPC typically outperforms KM in practice, as seen in the numerical results in Section 3.4.

Theorem 3.2. *Let $\mathcal{X}$ be generated according to the statistical data model described above. Then, under the clustering condition* (3.6)*, the partition $\hat{\mathcal{X}}_1,\dots,\hat{\mathcal{X}}_L$ of $\mathcal{X}$ inferred by KM corresponds to the true partition $\mathcal{X}_1,\dots,\mathcal{X}_L$ with probability at least $1-6N/M^2$.*

The proofs of Theorems 3.1 and 3.2 are provided in Appendix 3.A. We first note that the clustering condition (3.6) depends on a few model parameters only and all constants involved are explicit. Furthermore, the condition is identical for NNPC and KM, although the performance guarantee we obtain for KM (namely correct clustering) is stronger than that for NNPC (namely the NFC property). This is a consequence of both proofs relying on the same "separation condition" (namely (3.13) in Appendix 3.A) and the clustering condition (3.6) being sufficient for this separation condition to hold (see Appendix 3.A for further details).

Theorems 3.1 and 3.2 essentially state that NNPC and KM succeed even when the PSDs $s^{(\ell)}$ of the $X^{(\ell)}$ overlap significantly and the observations have missing entries and are contaminated by strong noise, all this provided that the observation length M is sufficiently large and the window function g is chosen to guarantee small $\mu_{\max}$. The clustering condition (3.6) suggests (recall that it is sufficient only) a tradeoff between the amount of overlap of pairs of PSDs $\{s^{(k)}, s^{(\ell)}\}$ (through $\min_{k,\ell\in[L]\colon k\neq\ell} d(X^{(k)}, X^{(\ell)})$), the observation length M, the sampling probability p, and the noise variance σ^2. It indicates, for example, that both algorithms tolerate shorter observation length M, more missing entries (i.e., smaller p), and stronger noise (i.e., larger σ^2) as the pairs $\{s^{(k)}, s^{(\ell)}\}$, $k \neq \ell$, overlap less and hence $\min_{k,\ell\in[L]\colon k\neq\ell} d(X^{(k)}, X^{(\ell)})$ is larger. Keeping σ^2 and p fixed, the first term on the RHS of (3.6), which accounts for the PSD estimation error owing to finite observation length M, vanishes as M becomes large. Since $d(X^{(k)}, X^{(\ell)}) \in [0, 1]$, we need $\mu_{\max} \ll 1$ to ensure that the clustering condition can be satisfied for finite M. To see how this can be accomplished, we consider $r^{(\ell)}$ of small effective support relative to M, i.e., $r^{(\ell)}[m] \approx 0$ for $m \geq M_0$ with $M_0 \ll M$, which is essentially equivalent to requiring that the $s^{(\ell)}$ be sufficiently smooth. We then choose g such that $g[m] \approx 1$ for $m \leq M_0$ and note that this ensures $h[m] \approx 1 - g[m](1 - |m|/M) \approx 1 - g[m] \approx 0$, for $m \leq M_0 \ll M$. Thanks to $\mu^{(\ell)} = \sum_{m=-\infty}^{\infty} |h[m]||r^{(\ell)}[m]| \approx \sum_{m=-M_0}^{M_0} |h[m]||r^{(\ell)}[m]| \ll 1$, we then get $\mu_{\max} \ll 1$. The clustering condition (3.6) can hence, indeed, be satisfied for finite M if the $r^{(\ell)}$ have small effective support. Note that the choice of g will affect the constant A (recall that $0 \leq g(f) \leq A < \infty$, $f \in [0, 1)$). Specifically, windows g of larger effective support have larger corresponding A in general.

To ensure high probability of success, we need to take $M \gg \sqrt{N}$, i.e., the observation length has to be large relative to the square root of the number of observations. We note that the results in Theorems 3.1 and 3.2 can easily be extended to colored noise processes, as long as the noise PSDs are identical for all $\ell \in [L]$.

We emphasize that the vast majority of analytical performance results for random process clustering available in the literature pertain to the asymptotic regime $M \to \infty$, with N fixed. The findings in (Kakizawa et al., 1998; Vilar and Pértega, 2004) are closest in spirit to ours and show that pairs of observations stemming from different generative models can be discriminated consistently (in the statistical sense), for $M \to \infty$, via a PSD-based distance measure, provided that the PSDs of all pairs of generative models differ on a set of positive Lebesgue measure.

Finally, we note that generalization of our analytical results to processes other than Gaussian such as, e.g., subgaussian processes, seems difficult as a version of the concentration inequality (Demanet et al., 2012, Lem. 1), upon which the proofs of Theorems 3.1 and 3.2 rely, does not appear to be available for non-Gaussian random vectors with dependent entries (see (Adamczak, 2015) for details). For i.i.d. subgaussian processes such an inequality was reported in (Rudelson and Vershynin, 2013); this is, however, not of interest here as i.i.d. processes have flat PSDs.

3.3. COMPARISON WITH THRESHOLDING-BASED SUBSPACE CLUSTERING

For finite observation length M, the random process clustering problem considered here can also be cast as a classical subspace clustering problem as studied in Chapter 2 simply by interpreting the observations $x_i^{(\ell)}$ as vectors $\mathbf{x}_i^{(\ell)} \in \mathbb{R}^M$. Numerical results, not reported here[1], demonstrate, however, that this approach leads to NNPC significantly outperforming its subspace clustering cousin, the TSC algorithm (Heckel and Bölcskei, 2015). Our next result, Proposition 3.1 below, provides analytical underpinning for this observation. Before stating the formal result, we develop some intuition. To this end, we consider

[1] but available at `http://www.nari.ee.ethz.ch/commth/research/`

statistically independent observations and set $p = 1$ (i.e., no missing entries). We then note that the clustering condition (3.6) for NNPC ensures that (using (3.17) and (3.18) in (3.13) together with (3.24) and (3.34), cf. Appendices 3.A and 3.B)

$$\mathrm{P}\Big[d(x_j^{(k)}, x_i^{(\ell)}) \leq d(x_v^{(\ell)}, x_i^{(\ell)})\Big] < \frac{6}{M^2}, \tag{3.7}$$

for j, $i \neq v$ and $k \neq \ell$. This guarantees that the probability of the NFC property being violated becomes small for M large, in particular, even when the PSD pairs $\{s^{(k)}, s^{(\ell)}\}$, $k \neq \ell$, overlap substantially and $\mathrm{SNR} := r^{(\ell)}[0]/\sigma^2 = 1/\sigma^2 < 1$, $\ell \in [L]$. For TSC (which constructs the sets $\mathcal{T}_i$ such that $|\langle \mathbf{x}_j, \mathbf{x}_i \rangle| \geq |\langle \mathbf{x}_v, \mathbf{x}_i \rangle|$ for all $j \in \mathcal{T}_i$ and all $v \notin \mathcal{T}_i$) applied to $\{\mathbf{x}_i^{(\ell)}\}_{i \in n_\ell, \ell \in [L]}$ the probability corresponding to the LHS of (3.7) is $\mathrm{P}[|\langle \mathbf{x}_j^{(k)}, \mathbf{x}_i^{(\ell)} \rangle| \geq |\langle \mathbf{x}_v^{(\ell)}, \mathbf{x}_i^{(\ell)} \rangle|]$. The next proposition establishes that this probability remains bounded away from 0 even when M grows large, unless the observations are noiseless and all the PSD pairs $\{s^{(k)}, s^{(\ell)}\}$, $k \neq \ell$, are supported on essentially disjoint frequency bands. These conditions are, however, hardly encountered in practice, and the corresponding clustering problem can be considered easy. The superior performance of NNPC as compared to TSC stems from the TSC similarity measure not exploiting the stationarity of the generative models. We proceed to the formal statement.

Proposition 3.1. *Let $x_i^{(\ell)}$ be a contiguous length-M observation of $\tilde{X}^{(\ell)}$ (note that we consider the case $p = 1$). Assume that the $x_i^{(\ell)}$ are independent across $\ell \in [L]$ and $i \in [n_\ell]$. Denote the vectors containing the elements of the $x_i^{(\ell)}$ by $\mathbf{x}_i^{(\ell)} \in \mathbb{R}^M$ and the corresponding covariance matrices by $\tilde{\mathbf{R}}^{(\ell)} := \mathbf{R}^{(\ell)} + \sigma^2 \mathbf{I}$, with $\mathbf{R}_{v,w}^{(\ell)} = r^{(\ell)}[w - v] = r^{(\ell)}[v - w]$, $\ell \in [L]$. Then, for $k \neq \ell$ and $v \neq i$, we have*

$$\mathrm{P}\Big[\Big|\Big\langle \mathbf{x}_j^{(k)}, \mathbf{x}_i^{(\ell)} \Big\rangle\Big| \geq \Big|\Big\langle \mathbf{x}_v^{(\ell)}, \mathbf{x}_i^{(\ell)} \Big\rangle\Big|\Big] \\ \geq \frac{1}{5\pi} \arctan\left(\frac{\sqrt{\operatorname{tr}\big(\tilde{\mathbf{R}}^{(k)} \tilde{\mathbf{R}}^{(\ell)}\big)}}{5\sqrt{3}\sqrt{\operatorname{tr}\big(\tilde{\mathbf{R}}^{(\ell)} \tilde{\mathbf{R}}^{(\ell)}\big)}} \right). \tag{3.8}$$

Proof: See Appendix 3.C.

Remark 3.1. *Note that, in contrast to Theorems 3.1 and 3.2, Proposition 3.1 assumes the observations to be statistically independent. This assumption turns out to be critical in the proof of Proposition 3.1.*

We next show, as announced, that the RHS of (3.8) remains strictly positive even when M grows large, unless the observations are noiseless and all pairs of PSDs have essentially disjoint support. To this end, we examine the behavior of $(1/M)\mathrm{tr}(\tilde{\mathbf{R}}^{(k)}\tilde{\mathbf{R}}^{(\ell)})$, $k \neq \ell$, and $(1/M)\mathrm{tr}(\tilde{\mathbf{R}}^{(\ell)}\tilde{\mathbf{R}}^{(\ell)})$ (the motivation for the normalization by M will become clear later). First note that $(1/M)\mathrm{tr}(\tilde{\mathbf{R}}^{(\ell)}\tilde{\mathbf{R}}^{(\ell)}) < \infty$ as $\sum_{m=-\infty}^{\infty}\left(\tilde{r}^{(\ell)}[m]\right)^2 < \infty$ by virtue of $\tilde{r}^{(\ell)} = r^{(\ell)}[m] + \sigma^2\delta[m] \in \ell_1$, which, in turn, follows from the assumption $r^{(\ell)} \in \ell_1$. The probability in (3.8) is hence bounded away from 0 unless $(1/M)\mathrm{tr}(\tilde{\mathbf{R}}^{(k)}\tilde{\mathbf{R}}^{(\ell)}) \approx 0$. It therefore remains to identify conditions for $(1/M)\mathrm{tr}(\tilde{\mathbf{R}}^{(k)}\tilde{\mathbf{R}}^{(\ell)}) \approx 0$ to hold. To this end, we note that

$$\begin{aligned}\frac{1}{M}\mathrm{tr}\left(\tilde{\mathbf{R}}^{(k)}\tilde{\mathbf{R}}^{(\ell)}\right) &= \frac{1}{M}\sum_{m=0}^{M-1}\sum_{n=0}^{M-1}\tilde{r}^{(k)}[n-m]\tilde{r}^{(\ell)}[n-m] \\ &= \sum_{m\in\mathcal{M}}\left(1-\frac{|m|}{M}\right)\tilde{r}^{(k)}[m]\tilde{r}^{(\ell)}[m] \qquad (3.9) \\ &= \int_0^1 (w * \tilde{s}^{(k)})(f)\tilde{s}^{(\ell)}(f)\mathrm{d}f, \qquad (3.10)\end{aligned}$$

where (3.9) is due to the Toeplitz structure and the symmetry of $\tilde{\mathbf{R}}^{(k)}$ and $\tilde{\mathbf{R}}^{(\ell)}$, (3.10) is by Parseval's Theorem, $\mathcal{M} := \{-M+1, -M+2, \ldots, M-1\}$, and $w(f) := \sum_{m\in\mathcal{M}}(1-|m|/M)e^{-\mathrm{i}2\pi fm} = \sin^2(\pi fM)/(M\sin^2(\pi f))$. As $w(f)$ is strictly positive on the interval $[0,1)$ (apart from its zeros which are supported on a set of measure 0) and the $\tilde{s}^{(\ell)}(f)$, $\ell \in [L]$, are non-negative, (3.10) is bounded away from 0 for finite M. As M grows large, $w(f)$ approaches the Dirac delta distribution, i.e., the "leakage" induced by w becomes small and we have (3.10) $\approx \int_0^1 \tilde{s}^{(k)}(f)\tilde{s}^{(\ell)}(f)\mathrm{d}f$. This integral vanishes for all $k \neq \ell$ if and only if $\sigma^2 = 0$ (recall that $\tilde{s}^{(\ell)}(f) = s^{(\ell)}(f) + \sigma^2$,

$\ell \in [L]$) and all pairs $\{s^{(k)}, s^{(\ell)}\}$, $k \neq \ell$, are supported on essentially disjoint frequency bands. This establishes the claim made above and concludes the argument.

3.4. NUMERICAL RESULTS[2]

We present numerical results for NNPC and KM on synthetic and on real data. In addition, we report results for KM followed by 100 k-means iterations (see the discussion in Section 3.1); this variant of KM will be referred to as iterated k-means (KMit). Furthermore, we compare NNPC, KM, and KMit with single linkage (SL), average linkage (AL), and complete linkage (CL) hierarchical clustering (Friedman et al., 2009, Sec. 14.3.12), all based on the L^1-distance measure (3.2). We also investigate variants of NNPC, KM, and KMit with the L^1-distance measure replaced by $d_{L^2}(x_i, x_j) := (\int_0^1 |\hat{s}_i(f) - \hat{s}_j(f)|^2 \, \mathrm{d}f)^{\frac{1}{2}}$, and variants of NNPC and KM with the L^1-distance measure replaced by $d_{L^\infty}(x_i, x_j) := \sup_{f \in [0,1)} |\hat{s}_i(f) - \hat{s}_j(f)|$ (we do not consider KMit here as d_{L^∞}-based k-means iterations do not seem sensible). NNPC and KM were implemented strictly according to the corresponding algorithm descriptions in Section 3.1. For SL, AL, and CL, we use the functions built into Matlab. Throughout, performance is measured in terms of the CE, i.e., the fraction of misclustered data points, also used in Chapter 2 (cf. the more formal definition in (2.11)). We report running times (excluding time for loading the data) obtained on a MacBook Pro with a 2.5 GHz Intel Core i7 CPU with 16 GB RAM.

3.4.1. Synthetic data

We investigate the tradeoff between the minimum distance $\min_{k,\ell \in [L]\colon k \neq \ell} d(X^{(k)}, X^{(\ell)})$, the observation length M, the sampling probability p, and the noise variance σ^2 as indicated by the clustering condition (3.6). Recall that the clustering condition is only sufficient

[2] Code to reproduce the experiments is available at `http://www.nari.ee.ethz.ch/commth/research/`.

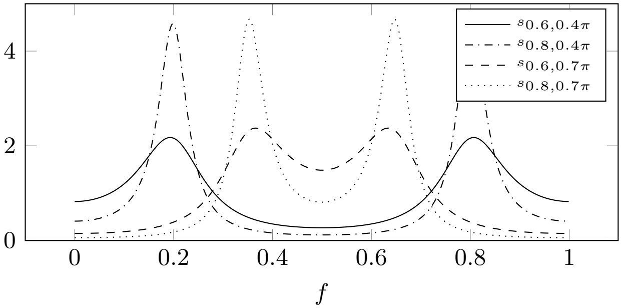


Fig. 3.1: Example PSDs of the form (3.11).

(and for NNPC guarantees the NFC property only). It is therefore unclear a priori to what extent the CE, indeed, follows the behavior indicated by the clustering condition.

We consider $L = 2$ second-order AR generative processes with PSDs of the form

$$s_{a,\nu}(f) = \frac{b^2(a,\nu)}{|1 - 2a\cos(\nu)e^{\mathrm{i}2\pi f} + a^2 e^{\mathrm{i}4\pi f}|^2}, \tag{3.11}$$

where $\nu \in [0, \pi]$, $a \in (0, 1)$, and $b^2(a, \nu) = 1/(\int_0^1 1/|1-2a\cos(\nu)e^{\mathrm{i}2\pi f} + a^2 e^{\mathrm{i}4\pi f}|^2 \mathrm{d}f)$ ensures that $\int_0^1 s_{a,\nu}(f)\mathrm{d}f = 1$. Figure 3.1 shows examples of $s_{a,\nu}(f)$ for different choices of a and ν. In the ensuing experiments, we set $s^{(1)}(f) = s_{0.6,0.7\pi}(f)$ and $s^{(2)}(f) = s_{0.6,\nu_2}(f)$, where ν_2 is variable and controls the locations of the peaks of $s^{(2)}$ and thereby the distance $d(X^{(1)}, X^{(2)})$. Indeed, varying ν_2 shifts the locations of the peaks of $s^{(2)}$ while essentially maintaining its shape. For the BT PSD estimator, we use a Bartlett window of length W defined as

$$g_W^B[m] := \begin{cases} 1 - |m|/\lfloor W/2 \rfloor, & \text{for } |m| \leq \lfloor W/2 \rfloor \\ 0, & \text{otherwise,} \end{cases} \tag{3.12}$$

and we set $W = 101$. Note that g_W^B satisfies the assumptions made about g in Section 3.2. The number of generative models $L = 2$ is assumed known throughout. The performance of NNPC is found (corresponding results are not shown here) to be rather insensitive to the choice of the parameter q as long as $10 \leq q \leq 25$; we set $q = 10$. For a given quadruple (ν_2, M, σ, p), a realization of the data set $\mathcal{X}$ is obtained by sampling $n = 25$ independent observations from $\check{X}^{(1)}$ and $\check{X}^{(2)}$ each, and the CE is estimated by averaging over 10 such independent realizations of $\mathcal{X}$. We do not normalize the BT PSD estimates to unit power.

Figure 3.2 shows that NNPC, KM, and KMit all exhibit roughly the same qualitative behavior as a function of $d(X^{(1)}, X^{(2)})$, M, $1/p$, and σ. In particular, for large enough $d(X^{(1)}, X^{(2)})$ all three algorithms yield a CE close to 0 even when σ^2 exceeds the signal power (i.e., when SNR < 1), when the observations have missing entries ($p < 1$), and when M is small. All three algorithms tolerate more noise and more missing entries as the observation length increases. These numerical results are in line with the *qualitative* tradeoff indicated by the (sufficient) clustering condition (3.6). The numerical constants in (4) are, however, too big for the clustering condition (4) to be sharp. NNPC consistently achieves the lowest CE, followed by KMit, and KM. The performance advantage of NNPC over KM and KMit can be attributed to the spectral clustering step, which leads to increased robustness to noise and missing entries. Finally, we note that KMit often yields a significantly lower CE than KM.

The results in Figure 3.4 indicate that the *qualitative* dependence of the CE on $d(X^{(1)}, X^{(2)})$, M, $1/p$, and σ for SL, AL, and CL is essentially identical to that for NNPC, KM, and KMit. For large σ and small $d(X^{(1)}, X^{(2)})$, M, or p, SL and AL lead, however, to a significantly larger CE than NNPC, KM, and KMit. The CE for CL is comparable to, but slightly larger than, that of KMit and significantly larger than that of NNPC.

Comparing the CE for NNPC, KM, and KMit in Figure 3.2 with that obtained for their d_{L^2} and d_{L^∞}-cousins in Figures 3.3 and 3.5, respectively, we note that, for all values of $d(X^{(1)}, X^{(2)})$, M, σ, and p the d_{L^2}-based variants of NNPC, KM, and KMit and the d_{L^∞}-based variants of NNPC and KM yield the same or larger CE

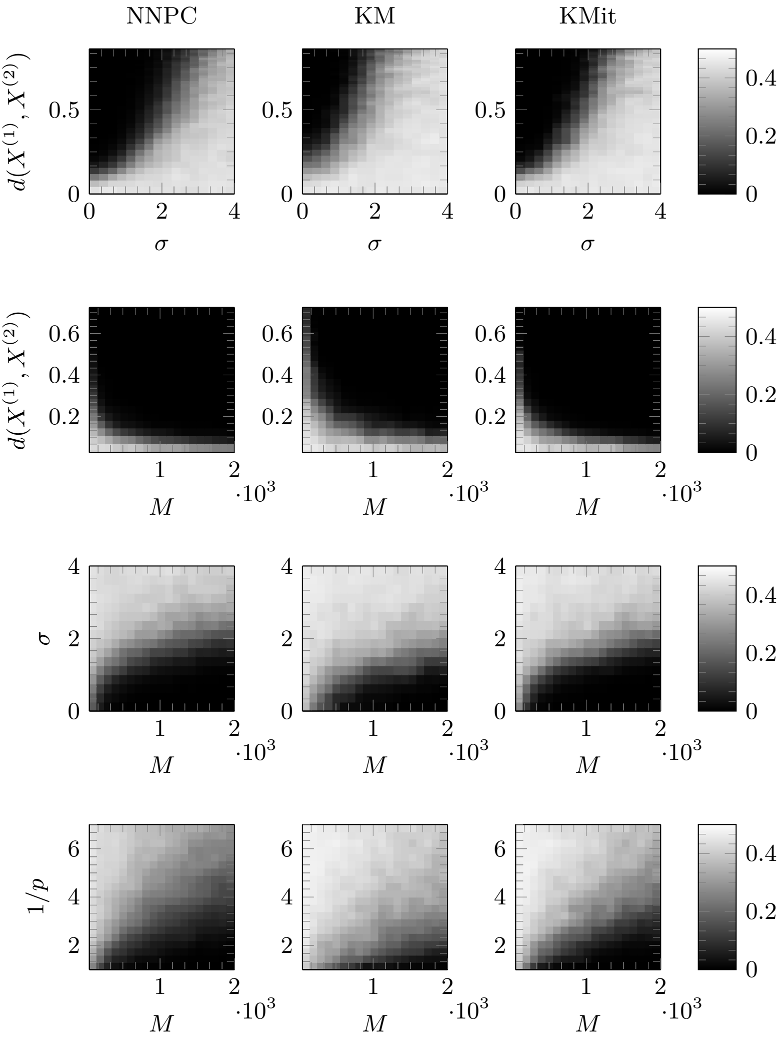


Fig. 3.2: Results of the synthetic data experiment. First row: CE as a function of σ and $d(X^{(1)}, X^{(2)})$ for $M = 400$ and $p = 1$. Second row: CE as a function of M and $d(X^{(1)}, X^{(2)})$ for $\sigma = 0.5$ and $p = 1$. Third row: CE as a function of M and σ for $\nu_2 = 0.62\pi$ ($d(X^{(1)}, X^{(2)}) \approx 0.2$) and $p = 1$. Bottom row: CE as a function of M and $1/p$ for $\nu_2 = 0.62\pi$ and $\sigma = 0.5$.

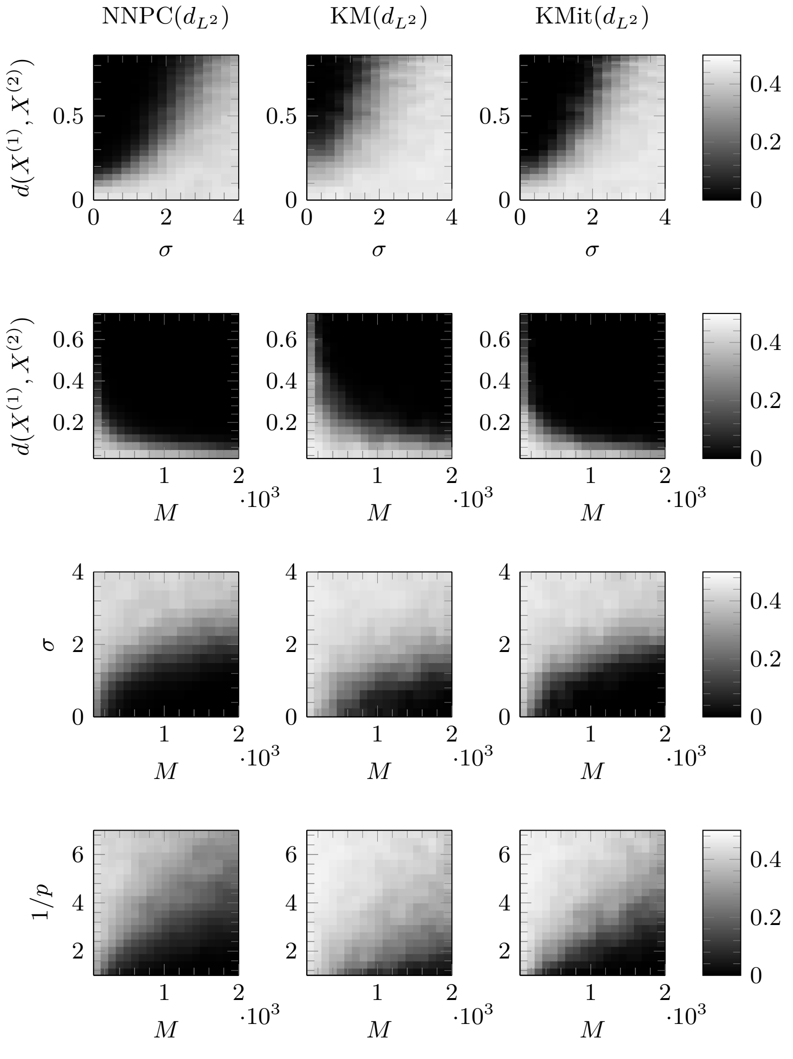


Fig. 3.3: CE as a function of $d(X^{(1)}, X^{(2)})$, M, σ, and p for variants of NNPC, KM, and KMit based on d_{L^2}, using the same values as in the setup in Figure 3.2 for the model parameters that are not varied.

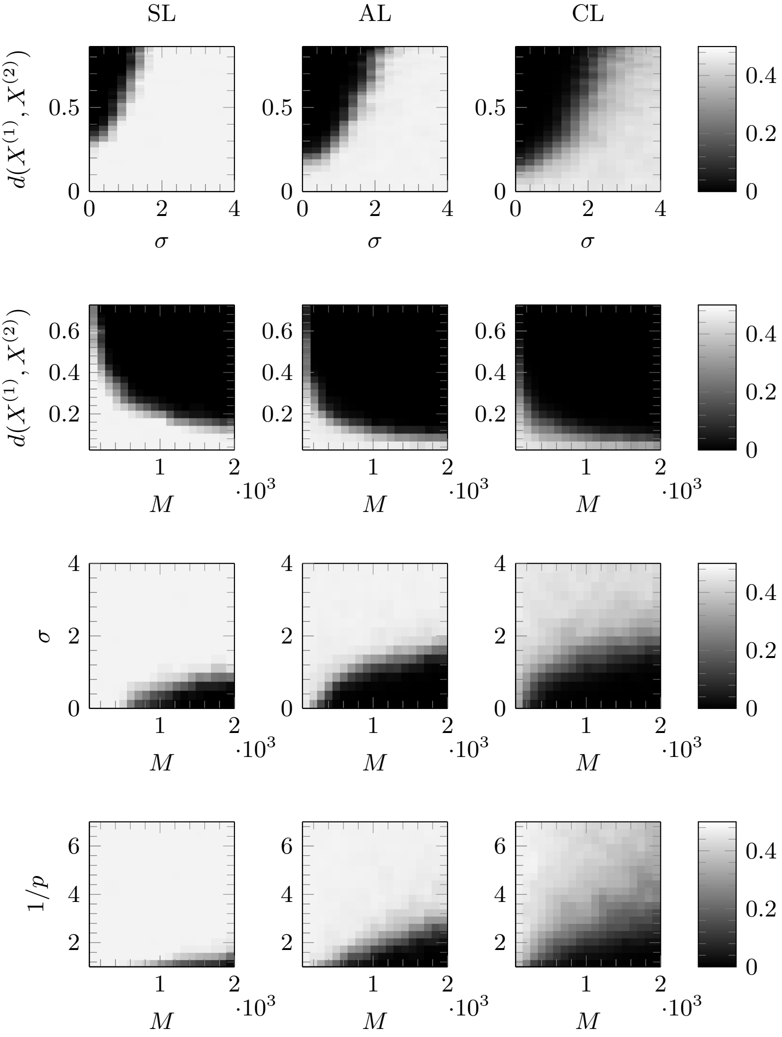


Fig. 3.4: CE for single linkage, average linkage, and complete linkage hierarchical clustering as a function of $d(X^{(1)}, X^{(2)})$, M, σ, and p, using the same values as in the setup in Figure 3.2 for the model parameters that are not varied.

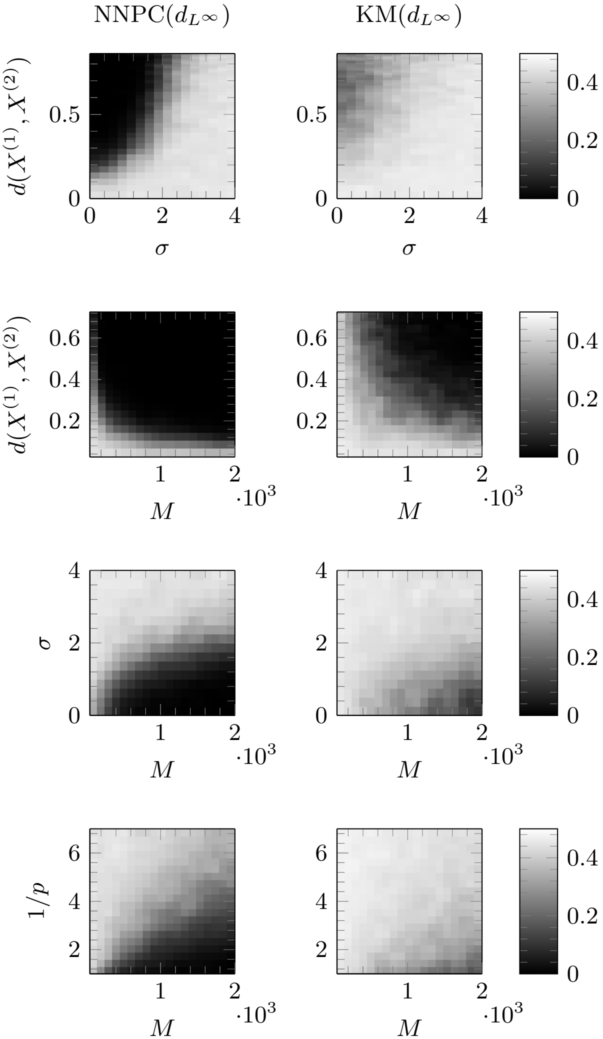


Fig. 3.5: CE as a function of $d(X^{(1)}, X^{(2)})$, M, σ, and p for variants of NNPC and KM based on d_{L^∞}, using the same values as in the setup in Figure 3.2 for the model parameters that are not varied.

than the respective original variants. This justifies usage of the L^1-based distance measure (3.2) also from a practical point of view. Finally, we note that normalizing the model PSDs (3.11) according to $(\int_0^1 s_{a,\nu}^2(f)\mathrm{d}f)^{\frac{1}{2}} = 1$ for d_{L^2} and $\sup_{f\in[0,1)} s_{a,\nu}(f) = 1$ for d_{L^∞} does not have a noticeable impact on the clustering performance.

3.4.2. Real data

We perform experiments on two data sets, namely on human motion data and on EEG data.

Human motion data: We consider the problem of clustering sequences of human motion data according to the underlying activities performed. Specifically, we consider the experiment conducted in (Li and Prakash, 2011; Khaleghi et al., 2012), which uses the Carnegie Mellon Motion Capture database[3] containing motion sequences of 149 subjects performing various activities. The clustering algorithm in (Li and Prakash, 2011) first fits a linear dynamical system model to each motion sequence and then performs standard k-means clustering with the estimated model parameters (organized into vectors) as data points. In (Khaleghi et al., 2012) an online clustering algorithm based on KM in combination with distributional distance is proposed. The motion vector-sequences in the Carnegie Mellon Motion Capture database describe the temporal evolution of marker positions on different body parts, recorded through optical tracking. The experiment in (Li and Prakash, 2011; Khaleghi et al., 2012) is based on subjects #16 and #35 for which the database contains 49 and 33 sequences, respectively, labeled either as "walking" or "running". We cluster the (scalar-valued) sequences describing the motion of the marker placed on the right foot of the subjects. It is argued in (Khaleghi et al., 2012) that these sequences can be considered stationary ergodic. We assume the number of generative models $L = 2$ to be known and set $q = 5$ (good performance was observed for $4 \leq q \leq 10$). For

[3] The Carnegie Mellon Motion Capture database is available at `http://mocap.cs.cmu.edu`.

Table 3.1: CE, S, and running time t (in seconds) for clustering of human motion sequences using NNPC, KM, and KMit as well as online clustering (OC) (Khaleghi et al., 2012, Algorithm 2) and complex linear dynamical systems (CLDS)-based clustering (Li and Prakash, 2011).

	NNPC			KM			KMit			OC	CLDS
subject	CE	S	t	CE	S	t	CE	S	t	S	S
#16	0.02	0.09	0.206	0.24	0.55	0.029	0.20	0.49	0.038	0.21	0.37
#35	0	0	0.185	0	0	0.017	0	0	0.024	0	0.10

the BT estimator, we use the Bartlett window g_W^B, defined in (3.12), with W given by the sequence length, and we normalize the BT PSD estimates to unit power. Table 3.1 lists the CE, the running times in seconds, and for comparison with the results in (Li and Prakash, 2011; Khaleghi et al., 2012) also the entropy S of the clustering confusion matrix (see (Li and Prakash, 2011, Sec. 6) for the definition of S). This comparison reveals that for subject #35 NNPC, KM, and KMit all outperform the algorithm in (Li and Prakash, 2011) and match the performance of that in (Khaleghi et al., 2012), while for subject #16 NNPC significantly outperforms both the algorithms in (Li and Prakash, 2011; Khaleghi et al., 2012) as well as KM and KMit.

EEG data: We perform an experiment similar to that in (Maharaj and D'Urso, 2011, Sec. 5), which considers clustering of segments of EEG recordings of healthy subjects and of subjects experiencing epileptic seizure according to whether seizure activity is present or not. It is argued in (Sanei and Chambers, 2008) that EEG recordings can be modeled as stationary ergodic random processes. We use subsets A and E of the publicly available[4] EEG data set described in (Andrzejak et al., 2001). Each of these two subsets contains 100 EEG segments of 23.6s duration, acquired at a sampling rate of 173.61Hz. We refer to

[4] The EEG recordings are available at `http://ntsa.upf.edu/downloads/andrzejak-rg-et-al-2001-indications-nonlinear-deterministic-and-finite-dimensional`.

(Andrzejak et al., 2001) for a more detailed description of acquisition and preprocessing aspects.

We compare the performance of NNPC, KM, and KMit as a function of W and q (for NNPC). We center each EEG segment by subtracting its (estimated) mean and use a Bartlett window g_W^B, as defined in (3.12), of variable length W for the BT PSD estimator. Furthermore, we normalize the PSD estimates to unit power and assume the number of clusters $L = 2$ to be known. Figure 3.6 shows the CE obtained for NNPC as a function of the window length W and of q, as well as the CE obtained for KM and KMit as a function of window length W. It can be seen that NNPC is robust to small variations of q and W around the pair (q, W) corresponding to the minimum CE (marked by a white dot in Figure 3.6). Similarly, KM and KMit yield a CE close to their respective minima for a large range of values for W. In Table 3.2, we report the minimum CE achieved by each algorithm, along with the corresponding running times and CE-minimizing values for W and q (in the case of NNPC), all chosen based on results depicted in Figure 3.6. The minimum CE obtained for NNPC is significantly lower than that corresponding to KM and KMit.

Table 3.2: Clustering EEG segments: Minimum CE, running time t (in seconds), and corresponding parameter choices.

	min CE	t	W	q
NNPC	0.005	0.694	840	3
KM	0.360	0.482	640	-
KMit	0.095	0.954	520	-

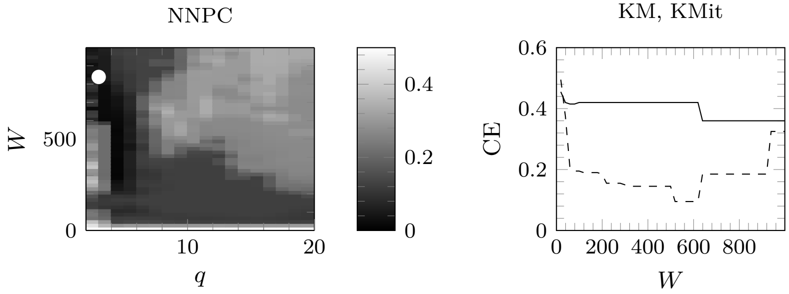


Fig. 3.6: Left: CE of NNPC for EEG recordings as a function of q and W. The white dot in the left figure shows the location of minimum CE. Right: CE of KM (solid line) and KMit (dashed line) as a function of W.

APPENDICES

3.A. PROOFS OF THEOREMS 3.1 AND 3.2

The central element in the proofs of Theorems 3.1 and 3.2 is the following result, proven in Appendix 3.B.

Theorem 3.3. *Consider a data set $\mathcal{X}$ generated according to the statistical data model described in Section 3.2. Then, the clustering condition* (3.6) *implies that*

$$\min_{\substack{k,\ell \in [L]:\\ k \neq \ell}} \min_{\substack{i \in [n_\ell],\\ j \in [n_k]}} d(x_j^{(k)}, x_i^{(\ell)}) > \max_{\ell \in [L]} \max_{\substack{i,j \in [n_\ell]:\\ i \neq j}} d(x_i^{(\ell)}, x_j^{(\ell)}) \tag{3.13}$$

holds with probability at least $1 - 6N/M^2$.

Theorem 3.3 says that under the clustering condition (3.6) observations stemming from the same generative model are closer (in terms of the distance measure d) than observations originating from different generative models. This property is known in the clustering literature as the *strict separation property* (Balcan et al., 2008). We now show how Theorems 3.1 and 3.2 follow directly from the strict separation property.

Proof of Theorem 3.1: Under the condition $q \leq \min_{\ell \in [L]}(n_\ell - 1)$ the NFC property is a direct consequence of (3.13), which by Theorem 3.3, is implied by the clustering condition (3.6). The condition $q \leq \min_{\ell \in [L]}(n_\ell - 1)$ is necessary for the NFC property to hold as choosing $q > \min_{\ell \in [L]}(n_\ell - 1)$ would force NNPC to select observations from $\mathcal{X} \backslash \mathcal{X}_\ell$ for at least one of the data points $x_i^{(\ell)}$, thereby resulting in a violation of the NFC property.

Proof of Theorem 3.2: The proof is effected by first showing that in Step 3 KM selects an observation with a different underlying generative model in every iteration, i.e., the set of cluster centers $\{x_{c_\ell}\}_{\ell=1}^L$ contains exactly one observation from each generative model, provided that the clustering condition (3.6) and hence, by Theorem 3.3, (3.13) holds. The argument is then concluded by noting that (3.13) implies directly that the partition $\hat{\mathcal{X}}_1, \dots, \hat{\mathcal{X}}_L$ obtained in Step 4 corresponds to the true partition $\mathcal{X}_1, \dots, \mathcal{X}_L$.

It remains to establish that the cluster centers x_{c_ℓ} selected in Step 3 of KM, indeed, all originate from different generative models. This is accomplished by induction. For $v = 1$ the claim holds trivially, as we have selected a single cluster center only, namely x_{c_1}. The base case is hence established. For the inductive step, suppose that after the vth iteration in Step 3 of KM the observations $\{x_{c_1}, \dots, x_{c_v}\}$ all come from different generative models, and assume w.l.o.g. that the generative model underlying x_{c_ℓ} has index ℓ, $\ell \in [v]$. In iteration $v + 1$ (i.e., for the selection of $x_{c_{v+1}}$), we have

$$\max_{i \in [N]} \min_{\ell \in [v]} d(x_i, x_{c_\ell}^{(\ell)})$$

$$= \max \left\{ \max_{\substack{k \in [v],\\ i \in [n_k]}} \min_{\ell \in [v]} d(x_i^{(k)}, x_{c_\ell}^{(\ell)}), \max_{\substack{k \in [L] \backslash [v],\\ i \in [n_k]}} \min_{\ell \in [v]} d(x_i^{(k)}, x_{c_\ell}^{(\ell)}) \right\}$$

$$= \max \left\{ \underbrace{\max_{\substack{k \in [v], \\ i \in [n_k]}} d(x_i^{(k)}, x_{c_k}^{(k)})}_{\leq \max\limits_{\ell \in [L]} \max\limits_{\substack{i,j \in [n_\ell]: \\ i \neq j}} d(x_i^{(\ell)}, x_j^{(\ell)})} , \underbrace{\max_{\substack{k \in [L]\setminus[v], \\ i \in [n_k]}} \min_{\ell \in [v]} d(x_i^{(k)}, x_{c_\ell}^{(\ell)})}_{\substack{\geq \min\limits_{\substack{k \in [L]\setminus[v], \\ i \in [n_k]}} \min\limits_{\ell \in [v]} d(x_i^{(k)}, x_{c_\ell}^{(\ell)}) \\ \geq \min\limits_{\substack{k, \ell \in [L]: \\ k \neq \ell}} \min\limits_{\substack{i \in [n_k], \\ j \in [n_\ell]}} d(x_i^{(k)}, x_j^{(\ell)})}} \right\} \tag{3.14}$$

$$= \max_{\substack{k \in [L]\setminus[v], \\ i \in [n_k]}} \min_{\ell \in [v]} d(x_i^{(k)}, x_{c_\ell}^{(\ell)}), \tag{3.15}$$

where we applied (3.13) to get (3.15) from (3.14). Note that in the maximization in (3.15) k runs over $[L]\setminus[v]$ (i.e., the maximization in (3.15) is over the observations in $\mathcal{X}\setminus(\mathcal{X}_1 \cup \cdots \cup \mathcal{X}_v)$), which implies that $x_{c_{v+1}}$ is guaranteed to correspond to a generative model that is different from those underlying $x_{c_1}, \ldots, x_{c_v}$. This completes the induction argument.

3.B. PROOF OF THEOREM 3.3

We start by quantifying the deviation of the estimated distances $d(x_j^{(k)}, x_i^{(\ell)})$ from the true distances $d(X^{(k)}, X^{(\ell)})$ due to the PSD estimation error caused by finite observation length, noise, and missing entries.

Let $\tilde{s}^{(\ell)}(f) \coloneqq s^{(\ell)}(f) + \sigma^2$, $f \in [0,1)$, $\ell \in [L]$, be the PSD of the noisy observation $\tilde{X}^{(\ell)}$ and denote the corresponding ACF by $\tilde{r}^{(\ell)}$. With $\hat{s}_i^{(\ell)}(f)$ as defined in (3.1) (recall that we use the modified window $\hat{g}[m] = g[m]/u[m]$ in the BT estimator (3.1)), set $e_i^{(\ell)}(f) \coloneqq \hat{s}_i^{(\ell)}(f) - \tilde{s}^{(\ell)}(f)$, and let $\varepsilon \coloneqq \max_{\ell \in [L], i \in [n_\ell]} \sup_{f \in [0,1)} |e_i^{(\ell)}(f)|$. We have for all $k, \ell \in [L]$, $j \in [n_k]$, $i \in [n_\ell]$,

$$d(x_j^{(k)}, x_i^{(\ell)}) = \frac{1}{2} \int_0^1 \left| \hat{s}_j^{(k)}(f) - \hat{s}_i^{(\ell)}(f) \right| \mathrm{d}f$$

$$= \frac{1}{2} \int_0^1 \left| s^{(k)}(f) + \sigma^2 + e_j^{(k)}(f) - (s^{(\ell)}(f) + \sigma^2 + e_i^{(\ell)}(f)) \right| \mathrm{d}f$$

$$\leq \frac{1}{2}\int_0^1 \left|s^{(k)}(f)-s^{(\ell)}(f)\right| \mathrm{d}f + \frac{1}{2}\int_0^1 \left|e_j^{(k)}(f) - e_i^{(\ell)}(f))\right| \mathrm{d}f$$

$$\leq d(X^{(k)}, X^{(\ell)}) + \frac{1}{2}\int_0^1 \left|e_j^{(k)}(f)\right| \mathrm{d}f + \frac{1}{2}\int_0^1 \left|e_i^{(\ell)}(f)\right| \mathrm{d}f \quad (3.16)$$

$$\leq d(X^{(k)}, X^{(\ell)}) + \varepsilon. \quad (3.17)$$

Applying the reverse triangle inequality, it follows similarly that

$$d(x_j^{(k)}, x_i^{(\ell)}) \geq d(X^{(k)}, X^{(\ell)}) - \varepsilon, \quad (3.18)$$

for all $k, \ell \in [L]$, $j \in [n_k]$, $i \in [n_\ell]$. Replacing the RHS of (3.13) by the upper bound in (3.17) and the LHS by the lower bound in (3.18), we find that (3.13) is implied by

$$\min_{k,\ell \in [L]\colon\, k \neq \ell} d(X^{(k)}, X^{(\ell)}) > 2\varepsilon. \quad (3.19)$$

We continue by upper-bounding ε. To this end, define $\mathbf{Q}_m \in \{0,1\}^{M\times M}$ according to $(\mathbf{Q}_m)_{u,v} = 1$, if $v - u = m$, and $(\mathbf{Q}_m)_{u,v} = 0$, else, and let $\hat{\mathbf{G}}(f) := \sum_{m\in\mathcal{M}} \hat{g}[m]\cos(2\pi f m)\mathbf{Q}_m$. Now, with $\mathbf{x} \in \mathbb{R}^M$ the random vector whose elements are given by $x_i^{(\ell)}$, it holds for $m \in \mathcal{M} = \{-M+1, -M+2, \ldots, M-1\}$ that

$$\hat{r}_i^{(\ell)}[m] = \frac{\mathbf{x}^\top \mathbf{Q}_m \mathbf{x}}{M} = \frac{\mathbf{y}^\top \mathbf{C}^\top \mathbf{P}_{\boldsymbol{\xi}}^\top \mathbf{Q}_m \mathbf{P}_{\boldsymbol{\xi}} \mathbf{C}\mathbf{y}}{M},$$

where we used $\mathbf{x} = \mathbf{P}_{\boldsymbol{\xi}}\mathbf{C}\mathbf{y}$, with the entries of $\mathbf{y}$ i.i.d. standard normal, $\mathbf{C} = (\mathbf{R} + \sigma^2\mathbf{I})^{1/2} \in \mathbb{R}^{M\times M}$ with $\mathbf{R}_{v,w} = r^{(\ell)}[w-v]$ the (Toeplitz) covariance matrix corresponding to M consecutive elements of $\tilde{X}^{(\ell)}$, and $\boldsymbol{\xi} \in \{0,1\}^M$ indicates the locations of the observed entries of $\mathbf{x}$. Note that $\mathbf{R}$ is identical for all contiguous length-M segments of $\tilde{X}^{(\ell)}$ thanks to stationarity, and $\mathbf{C}$ is symmetric because $\mathbf{R} + \sigma^2\mathbf{I}$ is symmetric. We next develop an upper bound on ε according to

$$\sup_{f\in[0,1)} \left|e_i^{(\ell)}(f)\right| = \sup_{f\in[0,1)} \left|\hat{s}_i^{(\ell)}(f) - \tilde{s}^{(\ell)}(f)\right|$$

$$
\begin{aligned}
&= \sup_{f \in [0,1)} \left| \sum_{m \in \mathcal{M}} \hat{g}[m] \hat{r}_i^{(\ell)}[m] e^{-\mathrm{i}2\pi f m} - \sum_{m \in \mathbb{Z}} \tilde{r}^{(\ell)}[m] e^{-\mathrm{i}2\pi f m} \right| \\
&= \sup_{f \in [0,1)} \Bigg| \underbrace{\sum_{m \in \mathcal{M}} \frac{\hat{g}[m]}{M} \left(\mathbf{x}^\top \mathbf{Q}_m \mathbf{x} - \mathbb{E}_{\mathbf{y}}\left[\mathbf{x}^\top \mathbf{Q}_m \mathbf{x}\right] \right) e^{-\mathrm{i}2\pi f m}}_{\substack{\frac{1}{M}\left(\mathbf{x}^\top \left(\sum_{m \in \mathcal{M}} \hat{g}[m] \cos(2\pi f m) \mathbf{Q}_m\right) \mathbf{x} \\ \qquad -\mathbb{E}_{\mathbf{y}}\left[\mathbf{x}^\top \left(\sum_{m \in \mathcal{M}} \hat{g}[m] \cos(2\pi f m) \mathbf{Q}_m\right) \mathbf{x}\right]\right)}} \\
&\quad + \sum_{m \in \mathcal{M}} \frac{\hat{g}[m]}{M} \mathbb{E}_{\mathbf{y}}\left[\mathbf{x}^\top \mathbf{Q}_m \mathbf{x}\right] e^{-\mathrm{i}2\pi f m} - \sum_{m \in \mathbb{Z}} \tilde{r}^{(\ell)}[m] e^{-\mathrm{i}2\pi f m} \Bigg| \\
&\leq \sup_{f \in [0,1)} \Bigg| \underbrace{\frac{1}{M} \Big(\mathbf{y}^\top \mathbf{C}^\top \mathbf{P}_{\boldsymbol{\xi}}^\top \hat{\mathbf{G}}(f) \mathbf{P}_{\boldsymbol{\xi}} \mathbf{C} \mathbf{y} - \mathbb{E}_{\mathbf{y}}\Big[\mathbf{y}^\top \mathbf{C}^\top \mathbf{P}_{\boldsymbol{\xi}}^\top \hat{\mathbf{G}}(f) \mathbf{P}_{\boldsymbol{\xi}} \mathbf{C} \mathbf{y}\Big] \Big)}_{=:\alpha_i^{(\ell)}(f)} \Bigg| \\
&\qquad + \Bigg| \underbrace{\sum_{m \in \mathcal{M}} \frac{\hat{g}[m]}{M} \mathbb{E}_{\mathbf{y}}\left[\mathbf{x}^\top \mathbf{Q}_m \mathbf{x}\right] - \sum_{m \in \mathbb{Z}} \tilde{r}^{(\ell)}[m]}_{=:\beta_i^{(\ell)}} \Bigg|, \qquad (3.20)
\end{aligned}
$$

where we used the fact that $\hat{g}[m](\mathbf{x}^\top \mathbf{Q}_m \mathbf{x} - \mathbb{E}_{\mathbf{y}}\left[\mathbf{x}^\top \mathbf{Q}_m \mathbf{x}\right])$ is a real-valued even sequence ($\hat{g}[m]$ and $\mathbf{x}^\top \mathbf{Q}_m \mathbf{x} = M \hat{r}_i^{(\ell)}[m]$ are real-valued even by definition, the latter property implies that $\mathbb{E}_{\mathbf{y}}\left[\mathbf{x}^\top \mathbf{Q}_m \mathbf{x}\right]$ is also real-valued and even). It now follows from (3.20) that

$$
\varepsilon \leq \max_{\ell \in [L], i \in [n_\ell]} \Big(\sup_{f \in [0,1)} \left|\alpha_i^{(\ell)}(f)\right| + \left|\beta_i^{(\ell)}\right| \Big)
$$

and hence (3.6) implies (3.13) via (3.19) on the event

$$
\mathcal{F}^\star := \bigcap_{\ell \in [L], i \in [n_\ell]} (\mathcal{F}_{1,i}^{(\ell)} \cap \mathcal{F}_{2,i}^{(\ell)})
$$

with

$$\mathcal{F}_{1,i}^{(\ell)} := \left\{ \sup_{f\in[0,1)} \left|\alpha_i^{(\ell)}(f)\right| < \frac{4A(B+\sigma^2)}{p^2}\sqrt{\frac{2\log M}{M}} \right\} \quad \text{and}$$

$$\mathcal{F}_{2,i}^{(\ell)} := \left\{ \left|\beta_i^{(\ell)}\right| < 8(1+p)\frac{A(1+\sigma^2)}{p^2}\sqrt{\frac{\log M}{M}} + \mu_{\max} \right\}.$$

With the upper bound on $\mathrm{P}[\bar{\mathcal{F}}_{1,i}^{(\ell)}]$ resulting from (3.24) and that on $\mathrm{P}[\bar{\mathcal{F}}_{2,i}^{(\ell)}]$ in (3.34), application of the union bound according to

$$\mathrm{P}[\mathcal{F}^\star] \geq 1 - \sum_{\ell\in[L],i\in[n_\ell]} \left(\mathrm{P}\left[\bar{\mathcal{F}}_{1,i}^{(\ell)}\right] + \mathrm{P}\left[\bar{\mathcal{F}}_{2,i}^{(\ell)}\right]\right) \geq 1 - \frac{6N}{M^2} \tag{3.21}$$

completes the proof.

We proceed to the upper bound on $\mathrm{P}[\bar{\mathcal{F}}_{1,i}^{(\ell)}]$.

Upper bound on $\mathrm{P}[\bar{\mathcal{F}}_{1,i}^{(\ell)}]$: Conditioning on $\boldsymbol{\xi}$ and setting $\mathbf{B} := \mathbf{C}^\top\mathbf{P}_{\boldsymbol{\xi}}^\top\hat{\mathbf{G}}(f)\mathbf{P}_{\boldsymbol{\xi}}\mathbf{C}$, we establish an upper bound on the tail probability of $\sup_{f\in[0,1)}\left|\alpha_i^{(\ell)}(f)\right|$ by invoking a well-known concentration of measure result for quadratic forms in Gaussian random vectors (Demanet et al., 2012, Lem. 1), namely

$$\begin{aligned}\mathrm{P}\Big[\,&\left|\mathbf{y}^\top\mathbf{B}\mathbf{y} - \mathbb{E}\left[\mathbf{y}^\top\mathbf{B}\mathbf{y}\right]\right| \\ &\geq \left\|\mathbf{B}+\mathbf{B}^\top\right\|_F\sqrt{\delta} + 2\|\mathbf{B}\|_{2\to 2}\delta \Big| \boldsymbol{\xi}\Big] \leq 2e^{-\delta}.\end{aligned} \tag{3.22}$$

Next, we note that $\|\mathbf{B}+\mathbf{B}^\top\|_F \leq 2\|\mathbf{B}\|_F \leq 2\sqrt{M}\|\mathbf{B}\|_{2\to 2}$ and

$$\begin{aligned}\|\mathbf{B}\|_{2\to 2} &\leq \|\underbrace{\mathbf{C}}_{=\mathbf{R}+\sigma^2\mathbf{I}}\|_{2\to 2}^2 \underbrace{\|\mathbf{P}_{\boldsymbol{\xi}}\|_{2\to 2}^2}_{\leq 1} \|\hat{\mathbf{G}}(f)\|_{2\to 2} \\ &\leq \frac{A(B+\sigma^2)}{p^2},\end{aligned}$$

where the second inequality follows as both $\mathbf{R}$ and $\hat{\mathbf{G}}(f)$ are symmetric Toeplitz matrices and hence, by (Gray, 2006, Lem. 4.1), $\|\mathbf{R}\|_{2\to 2} \leq$

$\sup_{f\in[0,1)} s^{(\ell)}(f) \leq B$ and

$$
\begin{aligned}
\|\hat{\mathbf{G}}(f)\|_{2\to 2} &\leq \sup_{f'\in[0,1)} \hat{g}(f') \\
&= \sup_{f'\in[0,1)} \frac{1}{p^2} g(f') + \underbrace{\left(\frac{1}{p} - \frac{1}{p^2}\right)}_{\leq 0} \underbrace{g[0]}_{=1} \\
&\leq \frac{1}{p^2} \sup_{f'\in[0,1)} g(f') = \frac{A}{p^2}, \qquad (3.23)
\end{aligned}
$$

where we used $\hat{g}[m] = (1/p^2)g[m] + (1/p - 1/p^2)g[0]\delta[m]$. Now, setting $\delta = 2\log(M)$ in (3.22) and using $\delta/M \leq \sqrt{\delta/M} < 1$, for $M \geq 1$, yields

$$
\begin{aligned}
\mathrm{P}\left[\bar{\mathcal{F}}_{1,i}^{(\ell)} \middle| \boldsymbol{\xi}\right] = \mathrm{P}\Bigg[\sup_{f\in[0,1)} \left|\alpha_i^{(\ell)}(f)\right| \\
\geq \frac{4A(B+\sigma^2)}{p^2} \sqrt{\frac{2\log M}{M}} \,\Bigg|\, \boldsymbol{\xi} \Bigg] \leq \frac{2}{M^2}. \qquad (3.24)
\end{aligned}
$$

The proof is concluded by noting that this bound holds uniformly over $\boldsymbol{\xi} \in \{0,1\}^M$ so that $\mathrm{P}[\bar{\mathcal{F}}_{1,i}^{(\ell)}] \leq 2/M^2$.

Upper bound on $\mathrm{P}[\bar{\mathcal{F}}_{2,i}^{(\ell)}]$: Setting $\tilde{\mathbf{G}} := \sum_{m\in\mathcal{M}} \hat{g}[m]\mathbf{Q}_m$, we start by rewriting the first sum in the definition of $\beta_i^{(\ell)}$ in (3.20) as

$$
\begin{aligned}
\sum_{m\in\mathcal{M}} \frac{\hat{g}[m]}{M} \mathbb{E}_{\mathbf{y}}\left[\mathbf{x}^\top \mathbf{Q}_m \mathbf{x}\right] &= \frac{1}{M} \mathbb{E}_{\mathbf{y}}\left[\mathbf{x}^\top \left(\sum_{m\in\mathcal{M}} \hat{g}[m]\mathbf{Q}_m\right)\mathbf{x}\right] \\
&= \frac{1}{M} \mathbb{E}_{\mathbf{y}}\left[\mathbf{y}^\top \mathbf{C}^\top \mathbf{P}_{\boldsymbol{\xi}}{}^\top \tilde{\mathbf{G}} \mathbf{P}_{\boldsymbol{\xi}} \mathbf{C} \mathbf{y}\right] \\
&= \frac{1}{M} \operatorname{tr}(\mathbf{C}^\top \mathbf{P}_{\boldsymbol{\xi}}{}^\top \tilde{\mathbf{G}} \mathbf{P}_{\boldsymbol{\xi}} \mathbf{C}) \\
&= \frac{1}{M} \operatorname{tr}(\mathbf{P}_{\boldsymbol{\xi}}{}^\top \tilde{\mathbf{G}} \mathbf{P}_{\boldsymbol{\xi}} \underbrace{\mathbf{C}\mathbf{C}^\top}_{=\mathbf{R}+\sigma^2\mathbf{I}}) \qquad (3.25)
\end{aligned}
$$

$$= \frac{1}{M} \sum_{u,v\in[M]} \xi_u \xi_v \tilde{\mathbf{G}}_{u,v} (\underbrace{\mathbf{R}_{v,u}}_{=\mathbf{R}_{u,v}} + \sigma^2 \delta[u-v]) \tag{3.26}$$

$$= \frac{1}{M} \boldsymbol{\xi}^\top (\tilde{\mathbf{G}} \odot (\mathbf{R} + \sigma^2 \mathbf{I})) \boldsymbol{\xi}. \tag{3.27}$$

Now, setting $\mathbf{D} \coloneqq \tilde{\mathbf{G}} \odot (\mathbf{R} + \sigma^2 \mathbf{I})$ and using (3.27), we have

$$\begin{aligned} &\left|\beta_i^{(\ell)}\right| \\ &= \left| \frac{1}{M} \boldsymbol{\xi}^\top \mathbf{D} \boldsymbol{\xi} - \frac{1}{M} \mathbb{E}[\boldsymbol{\xi}^\top \mathbf{D} \boldsymbol{\xi}] + \frac{1}{M} \mathbb{E}[\boldsymbol{\xi}^\top \mathbf{D} \boldsymbol{\xi}] - \sum_{m\in\mathbb{Z}} \tilde{r}^{(\ell)}[m] \right| \\ &\leq \left| \frac{1}{M} \big(\boldsymbol{\xi}^\top \mathbf{D} \boldsymbol{\xi} - \mathbb{E}[\boldsymbol{\xi}^\top \mathbf{D} \boldsymbol{\xi}]\big) \right| + \left| \frac{1}{M} \mathbb{E}[\boldsymbol{\xi}^\top \mathbf{D} \boldsymbol{\xi}] - \sum_{m\in\mathbb{Z}} \tilde{r}^{(\ell)}[m] \right| \qquad (3.28) \\ &\leq \left| \underbrace{\frac{1}{M} \big(\boldsymbol{\xi}^\top \mathbf{D} \boldsymbol{\xi} - \mathbb{E}[\boldsymbol{\xi}^\top \mathbf{D} \boldsymbol{\xi}]\big)}_{=:\gamma_i^{(\ell)}} \right| + \mu_{\max}. \qquad (3.29) \end{aligned}$$

Here, the last inequality is a consequence of the following upper bound on the second term in (3.28)

$$\begin{aligned} &\left| \frac{1}{M} \mathbb{E}[\boldsymbol{\xi}^\top \mathbf{D} \boldsymbol{\xi}] - \sum_{m\in\mathbb{Z}} \tilde{r}^{(\ell)}[m] \right| \\ &= \left| \frac{1}{M} \sum_{v,w\in[M]} \mathbb{E}[\xi_v \xi_w]\, \tilde{\mathbf{G}}_{v,w} (\mathbf{R}_{v,w} + \sigma^2 \delta[v-w]) \right. \\ &\qquad \left. - \sum_{m\in\mathbb{Z}} \tilde{r}^{(\ell)}[m] \right| \qquad (3.30) \\ &= \left| \frac{1}{M} \sum_{v,w\in[M]} \underbrace{\mathbb{E}[\xi_v \xi_w]\, \hat{g}[v-w]}_{u[v-w]\hat{g}[v-w]=g[v-w]}\ \tilde{r}^{(\ell)}[v-w] - \sum_{m\in\mathbb{Z}} \tilde{r}^{(\ell)}[m] \right| \\ &= \left| \underbrace{g[0]}_{=1} (r^{(\ell)}[0] + \sigma^2) - (r^{(\ell)}[0] + \sigma^2) \right. \end{aligned}$$

$$
\begin{aligned}
&\qquad + \sum_{m \in \mathcal{M}\setminus\{0\}} \left(\frac{M - |m|}{M} g[m] r^{(\ell)}[m] - r^{(\ell)}[m] \right) \\
&\qquad - \sum_{m \in \mathbb{Z}\setminus\mathcal{M}} r^{(\ell)}[m] \Bigg| \qquad (3.31) \\
&\leq \sum_{m \in \mathbb{Z}} |h[m]| |r^{(\ell)}[m]| \\
&\leq \mu_{\max}, \qquad (3.32)
\end{aligned}
$$

where (3.30) follows from the equality (3.27)=(3.26) and from $\tilde{r}^{(\ell)}[m] = r^{(\ell)}[m] + \sigma^2 \delta[m]$. We continue by establishing a bound on the tail probability of $|\gamma_i^{(\ell)}|$. To this end, we note that

$$
\begin{aligned}
\|\mathbf{D}\|_{2\to2} &= \|\tilde{\mathbf{G}} \odot (\mathbf{R} + \sigma^2 \mathbf{I})\|_{2\to2} \\
&= (1+\sigma^2) \left\| \tilde{\mathbf{G}} \odot \left(\frac{\mathbf{R} + \sigma^2 \mathbf{I}}{1+\sigma^2} \right) \right\|_{2\to2} \\
&\leq (1+\sigma^2) \|\tilde{\mathbf{G}}\|_{2\to2} \\
&\leq \frac{A(1+\sigma^2)}{p^2}, \qquad (3.33)
\end{aligned}
$$

where we used the fact that $(\mathbf{R} + \sigma^2 \mathbf{I})/(1+\sigma^2)$ is a symmetric positive semi-definite matrix with ones on its main diagonal, and we employed (Horn and Johnson, 1991, Thm. 5.5.11) in the first inequality, and steps analogous to those in (3.23) to obtain the second inequality.

Now, using (3.29) we get

$$
\begin{aligned}
\mathrm{P}\left[\bar{\mathcal{F}}_{2,i}^{(\ell)}\right] &\leq \mathrm{P}\left[\left|\gamma_i^{(\ell)}\right| \geq 8(1+p) \frac{A(1+\sigma^2)}{p^2} \sqrt{\frac{\log M}{M}} \right] \\
&< \mathrm{P}\left[\left|\gamma_i^{(\ell)}\right| > 8(1+p) \|\mathbf{D}\|_{2\to2} \sqrt{\frac{\log M}{M}} \right] < \frac{4}{M^2}, \qquad (3.34)
\end{aligned}
$$

where the second inequality follows from the upper bound on $\|\mathbf{D}\|_{2\to2}$ in (3.33) and the third inequality is an application of Lemma 3.1 with $\mathbf{H} := \mathbf{D}$ and $t := 8(1+p)\|\mathbf{D}\|_{2\to2}\sqrt{M \log M}$.

A final remark concerns the concentration inequality for quadratic forms in Boolean random vectors reported in the following Lemma 3.1. Such concentration inequalities, or more generally, concentration inequalities for multivariate polynomials of boolean random variables have been studied extensively in the context of random graph theory (Schudy and Sviridenko, 2012). Unfortunately, the bounds available in the literature typically come in terms of functions of the entries of $\mathbf{H}$ that do not lead to crisp statements in the context of the process clustering problem considered here. We therefore develop a new concentration result in Lemma 3.1, which depends on $\|\mathbf{H}\|_{2\to 2}$ only. The proof of this result is based on techniques developed in (Rudelson and Vershynin, 2013).

Lemma 3.1. *Let* $\mathbf{H} \in \mathbb{R}^{M\times M}$ *be a (deterministic) symmetric matrix and let* $\boldsymbol{\xi} \in \{0,1\}^M$ *be a random vector with i.i.d. Bernoulli entries drawn according to* $\mathrm{P}[\xi_i = 1] = 1 - \mathrm{P}[\xi_i = 0] = p$, $i \in [M]$*. Then, we have*

$$\begin{aligned}\mathrm{P}\big[\big|\boldsymbol{\xi}^\top \mathbf{H}\boldsymbol{\xi} - \mathbb{E}[\boldsymbol{\xi}^\top \mathbf{H}\boldsymbol{\xi}]\big| > t\big]& \\ < 4\exp\left(-\frac{t^2}{32(1+p)^2 M\|\mathbf{H}\|_{2\to 2}^2}\right).& \qquad (3.35)\end{aligned}$$

Proof. The proof is effected by adapting the proof of (Rudelson and Vershynin, 2013, Thm. 1.1), which provides a concentration inequality for quadratic forms in zero-mean subgaussian random vectors. We start by decomposing $\boldsymbol{\xi}^\top \mathbf{H}\boldsymbol{\xi} - \mathbb{E}\big[\boldsymbol{\xi}^\top \mathbf{H}\boldsymbol{\xi}\big]$ according to

$$\begin{aligned}&\boldsymbol{\xi}^\top \mathbf{H}\boldsymbol{\xi} - \mathbb{E}\big[\boldsymbol{\xi}^\top \mathbf{H}\boldsymbol{\xi}\big] \\ &\quad = \sum_{i\in[M]} \mathbf{H}_{i,i}(\xi_i^2 - \mathbb{E}\big[\xi_i^2\big]) + \sum_{\substack{i,j\in[M]:\\ i\neq j}} \mathbf{H}_{i,j}(\xi_i\xi_j - \mathbb{E}[\xi_i\xi_j]) \\ &\quad = \underbrace{\sum_{i\in[M]} \mathbf{H}_{i,i}(\xi_i - p)}_{=:S_{\mathrm{diag}}} + \underbrace{\sum_{\substack{i,j\in[M]:\\ i\neq j}} \mathbf{H}_{i,j}(\xi_i\xi_j - p^2)}_{=:S_{\mathrm{off}}},\end{aligned}$$

where we used the fact that the ξ_i, $i \in [M]$, are $\{0,1\}$-valued and statistically independent. Now, we have

$$\begin{aligned}
\mathrm{P}\big[\big|\boldsymbol{\xi}^\top \mathbf{H}\boldsymbol{\xi} - \mathbb{E}\big[\boldsymbol{\xi}^\top \mathbf{H}\boldsymbol{\xi}\big]\big| > t\big] &\le \mathrm{P}[|S_{\mathrm{diag}}| + |S_{\mathrm{off}}| > t] \\
&\le \mathrm{P}[|S_{\mathrm{diag}}| > t/2] + \mathrm{P}[|S_{\mathrm{off}}| > t/2] \qquad (3.36)\\
&\le 2\exp\left(-\frac{2t^2}{\sum_{i\in[M]}(\mathbf{H}_{i,i})^2}\right) \\
&\qquad + 2\exp\left(-\frac{t^2}{32(1+p)^2 M\|\mathbf{H}\|_{2\to 2}^2}\right) \qquad (3.37)\\
&< 4\exp\left(-\frac{t^2}{32(1+p)^2 M\|\mathbf{H}\|_{2\to 2}^2}\right),
\end{aligned}$$

where (3.37) follows from the upper bounds on $\mathrm{P}[|S_{\mathrm{diag}}| > t/2]$ and $\mathrm{P}[|S_{\mathrm{off}}| > t/2]$ established below, and the last inequality is thanks to $\sum_{i\in[M]}(\mathbf{H}_{i,i})^2 \le M \max_{i\in[M]}(\mathbf{H}_{i,i})^2 \le M\|\mathbf{H}\|_{2\to 2}^2$ obtained from $\|\mathbf{H}\|_{2\to 2}^2 = \max_{\|\mathbf{x}\|_2=1}\|\mathbf{H}\mathbf{x}\|_2^2 \ge \max_{\|\mathbf{x}\|_2=1,\mathbf{x}\in\{0,1\}^M}\|\mathbf{H}\mathbf{x}\|_2^2 = \max_{i\in[M]} \sum_{j\in[M]}(\mathbf{H}_{j,i})^2 \ge \max_{i\in[M]}(\mathbf{H}_{i,i})^2$.

Upper bound on $\mathrm{P}[|S_{\mathrm{diag}}| > t/2]$: Note that the $\mathbf{H}_{i,i}(\xi_i - p)$, $i \in [M]$, are independent, bounded, zero-mean random variables with $a_i \le \mathbf{H}_{i,i}(\xi_i - p) \le b_i$, $a_i, b_i \in \mathbb{R}$, $i \in [M]$. We can therefore apply Hoeffding's inequality (Boucheron et al., 2013, Thm. 2.8), which upon noting that $(b_i - a_i)^2 = \mathbf{H}_{i,i}^2$ yields

$$\mathrm{P}[|S_{\mathrm{diag}}| > t/2] < 2\exp\left(-\frac{2t^2}{\sum_{i\in[M]}(\mathbf{H}_{i,i})^2}\right).$$

Upper bound on $\mathrm{P}[|S_{\mathrm{off}}| > t/2]$: We start by decoupling (Foucart and Rauhut, 2013, Sec. 8.4) the sum S_{off} over the off-diagonal entries of $\mathbf{H}$, then upper-bound the moment generating function of S_{off}, and use the resulting upper bound to get an upper bound on $\mathrm{P}[S_{\mathrm{off}} > t/2]$ via the exponential Chebyshev inequality. The final result follows by noting that $\mathrm{P}[S_{\mathrm{off}} > t/2] = \mathrm{P}[S_{\mathrm{off}} < -t/2]$ and applying the union bound.

To decouple S_{off}, consider i.i.d. Bernoulli random variables $\nu_i \in \{0,1\}$, $i \in [M]$, with $\mathrm{P}[\nu_i = 0] = \mathrm{P}[\nu_i = 1] = 1/2$, and set $\boldsymbol{\nu} = [\nu_1 \ \dots \ \nu_M]^\top$. With

$$S_\nu \coloneqq \sum_{i,j \in [M]} \nu_i (1-\nu_j) \mathbf{H}_{i,j} (\xi_i - p)(\xi_j + p),$$

we have $S_{\text{off}} = 4\mathbb{E}_{\boldsymbol{\nu}}[S_\nu]$ thanks to the symmetry of $\mathbf{H}$ (i.e., $\mathbf{H}_{i,j} = \mathbf{H}_{j,i}$), and $\mathbb{E}[\nu_i(1-\nu_j)] = 1/4$, for $i \neq j$, and $\mathbb{E}[\nu_i(1-\nu_j)] = 0$, for $i = j$. Setting $\mathcal{I}_\nu \coloneqq \{i \in [M] \colon \nu_i = 1\}$, we can express S_ν as

$$\begin{aligned} S_\nu &= \sum_{i \in \mathcal{I}_\nu, j \in \overline{\mathcal{I}_\nu}} \mathbf{H}_{i,j} (\xi_i - p)(\xi_j + p) \\ &= \sum_{i \in \mathcal{I}_\nu} (\xi_i - p) \Bigg(\sum_{j \in \overline{\mathcal{I}_\nu}} \mathbf{H}_{i,j} (\xi_j + p) \Bigg). \end{aligned} \tag{3.38}$$

We continue by upper-bounding the moment generating function of S_{off} via Jensen's inequality according to

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\xi}}[\exp(\lambda S_{\text{off}})] &= \mathbb{E}_{\boldsymbol{\xi}}[\exp(\lambda 4 \mathbb{E}_{\boldsymbol{\nu}}[S_\nu])] \\ &\leq \mathbb{E}_{\boldsymbol{\xi},\boldsymbol{\nu}}[\exp(4\lambda S_\nu)], \end{aligned} \tag{3.39}$$

where $\lambda > 0$ is a deterministic parameter. It follows from (3.38) that S_ν, conditioned on $\boldsymbol{\nu}$ and on the ξ_j, with $j \in \overline{\mathcal{I}_\nu}$, is a linear combination of independent bounded zero-mean random variables. We therefore have

$$\begin{aligned} &\mathbb{E}_{\xi_i, i \in \mathcal{I}_\nu}[\exp(4\lambda S_\nu)] \\ &= \mathbb{E}_{\xi_i, i \in \mathcal{I}_\nu}\left[\exp\left(4\lambda \sum_{i \in \mathcal{I}_\nu} (\xi_i - p) \Bigg(\sum_{j \in \overline{\mathcal{I}_\nu}} \mathbf{H}_{i,j}(\xi_j + p) \Bigg)\right)\right] \\ &= \prod_{i \in \mathcal{I}_\nu} \mathbb{E}_{\xi_i}\left[\exp\left(4\lambda (\xi_i - p) \Bigg(\sum_{j \in \overline{\mathcal{I}_\nu}} \mathbf{H}_{i,j}(\xi_j + p) \Bigg)\right)\right] \end{aligned} \tag{3.40}$$

$$
\begin{aligned}
&\leq \prod_{i \in \mathcal{I}_\nu} \exp\left(2\lambda^2 \Big(\sum_{j \in \overline{\mathcal{I}_\nu}} \mathbf{H}_{i,j}(\xi_j + p) \Big)^2 \right) && (3.41)\\
&= \exp\left(2\lambda^2 \sum_{i \in \mathcal{I}_\nu} \underbrace{\Big(\sum_{j \in \overline{\mathcal{I}_\nu}} \mathbf{H}_{i,j}(\xi_j + p) \Big)^2}_{=\mathbf{H}_i(\mathbf{I}-\mathbf{P}_{\boldsymbol{\nu}})(\boldsymbol{\xi}+p\mathbf{1})} \right)\\
&= \exp\Big(2\lambda^2 \|\mathbf{P}_{\boldsymbol{\nu}}\mathbf{H}(\mathbf{I} - \mathbf{P}_{\boldsymbol{\nu}})(\boldsymbol{\xi} + p\mathbf{1})\|_2^2\Big)\\
&\leq \exp\left(2\lambda^2 \underbrace{\|\mathbf{P}_{\boldsymbol{\nu}}\|_{2\to 2}^2}_{\leq 1} \|\mathbf{H}\|_{2\to 2}^2 \underbrace{\|\mathbf{I} - \mathbf{P}_{\boldsymbol{\nu}}\|_{2\to 2}^2}_{\leq 1} \underbrace{\|\boldsymbol{\xi} + p\mathbf{1}\|_2^2}_{\leq M(1+p)^2} \right)\\
&\leq \exp\Big(2\lambda^2 (1+p)^2 M \|\mathbf{H}\|_{2\to 2}^2\Big), && (3.42)
\end{aligned}
$$

where we used the independence of the ξ_i, $i \in \mathcal{I}_\nu$, to get (3.40), and Hoeffding's Lemma in the step leading from (3.40) to (3.41). Note that instead of Hoeffding's Lemma we could also apply (Buldygin and Moskvichova, 2013, Thm. 2.1) to get a sharper bound on (3.40), but this would not lead to a different scaling behavior of (3.35) in terms of p or M.

Combining (3.42) with (3.39) and noting that the bound (3.42) does not depend on $\boldsymbol{\nu}$ and ξ_j, $j \in \overline{\mathcal{I}_\nu}$, it follows that

$$
\begin{aligned}
\mathbb{E}_{\boldsymbol{\xi},\boldsymbol{\nu}}[\exp(\lambda S_{\text{off}})] &\leq \mathbb{E}_{\boldsymbol{\xi},\boldsymbol{\nu}}[\exp(4\lambda S_\nu)]\\
&= \mathbb{E}_{\boldsymbol{\nu}}\Big[\mathbb{E}_{\xi_j, j \in \overline{\mathcal{I}_\nu}}[\mathbb{E}_{\xi_i, i \in \mathcal{I}_\nu}[\exp(4\lambda S_\nu)]]\Big]\\
&\leq \mathbb{E}_{\boldsymbol{\nu}}\Big[\mathbb{E}_{\xi_j, j \in \overline{\mathcal{I}_\nu}}\Big[\exp\Big(2\lambda^2(1+p)^2 M \|\mathbf{H}\|_{2\to 2}^2\Big)\Big]\Big]\\
&= \exp\Big(2\lambda^2(1+p)^2 M \|\mathbf{H}\|_{2\to 2}^2\Big). && (3.43)
\end{aligned}
$$

We finally use (3.43) and the exponential Chebyshev inequality to get the upper bound

$$
\mathrm{P}[S_{\text{off}} > t/2] \leq \exp\Big(-\lambda t/2 + 2\lambda^2(1+p)^2 M \|\mathbf{H}\|_{2\to 2}^2\Big), \qquad (3.44)
$$

which holds for all $\lambda > 0$. Minimizing (3.44) over $\lambda > 0$ yields

$$\mathrm{P}[S_{\text{off}} > t/2] \leq \exp\left(-\frac{t^2}{32(1+p)^2 M \|\mathbf{H}\|_{2\to 2}^2}\right). \tag{3.45}$$

□

3.C. PROOF OF PROPOSITION 3.1

Recall that $\mathbf{x}_i^{(\ell)} = \mathbf{C}^{(\ell)}\mathbf{y}_i^{(\ell)}$, $\ell \in [L]$, $i \in [n_\ell]$, where $\mathbf{y}_i^{(\ell)}$ is an i.i.d. standard normal random vector and $\mathbf{C}^{(\ell)} := (\tilde{\mathbf{R}}^{(\ell)})^{1/2}$. Setting $\sigma^{(k,\ell)} := \|\mathbf{C}^{(k)^\top}\mathbf{C}^{(\ell)}\mathbf{y}_i^{(\ell)}\|_2$, conditional on $\mathbf{y}_i^{(\ell)}$, $\langle \mathbf{x}_j^{(k)}, \mathbf{x}_i^{(\ell)} \rangle$ and $\langle \mathbf{x}_v^{(\ell)}, \mathbf{x}_i^{(\ell)} \rangle$, for $k \neq \ell$ and $v \neq i$, are independent (as a consequence of the mutual independence of the $\mathbf{x}_i^{(\ell)}$, $\ell \in [L]$, $i \in [n_\ell]$, which is by assumption) and distributed according to $\mathcal{N}(0, {\sigma^{(k,\ell)}}^2)$ and $\mathcal{N}(0, {\sigma^{(\ell,\ell)}}^2)$, respectively. Conditional on $\mathbf{y}_i^{(\ell)}$, or equivalently, conditional on $\sigma^{(k,\ell)}$ and $\sigma^{(\ell,\ell)}$, $|\langle \mathbf{x}_j^{(k)}, \mathbf{x}_i^{(\ell)} \rangle|$ and $|\langle \mathbf{x}_v^{(\ell)}, \mathbf{x}_i^{(\ell)} \rangle|$ hence have half-normal distributions and we get

$$\begin{aligned}
&\mathrm{P}\left[\left|\left\langle \mathbf{x}_j^{(k)}, \mathbf{x}_i^{(\ell)} \right\rangle\right| < \left|\left\langle \mathbf{x}_v^{(\ell)}, \mathbf{x}_i^{(\ell)} \right\rangle\right| \frac{\sigma^{(k,\ell)}}{\sigma^{(\ell,\ell)}}\right] \\
&\quad = \int_0^\infty \frac{\sqrt{2}}{\sigma^{(\ell,\ell)}\sqrt{\pi}} e^{-\frac{x^2}{2{\sigma^{(\ell,\ell)}}^2}} \int_0^x \frac{\sqrt{2}}{\sigma^{(k,\ell)}\sqrt{\pi}} e^{-\frac{y^2}{2{\sigma^{(k,\ell)}}^2}} \,\mathrm{d}y \,\mathrm{d}x \\
&\quad = \int_0^\infty \frac{\sqrt{2}}{\sigma^{(\ell,\ell)}\sqrt{\pi}} e^{-\frac{x^2}{2{\sigma^{(\ell,\ell)}}^2}} \operatorname{erf}\left(\frac{x}{\sigma^{(k,\ell)}\sqrt{2}}\right) \mathrm{d}x \\
&\quad = 1 - \frac{2}{\pi}\arctan\left(\frac{\sigma^{(k,\ell)}}{\sigma^{(\ell,\ell)}}\right),
\end{aligned} \tag{3.46}$$

where we used the integral formula (Ng and Geller, 1969, Eqn. 2, p. 7) $\int_0^\infty \operatorname{erf}(ax)e^{-b^2x^2}\mathrm{d}x = (\pi/2 - \arctan(b/a))/(b\sqrt{\pi})$, with $a = 1/(\sigma^{(k,\ell)}\sqrt{2})$ and $b = 1/(\sigma^{(\ell,\ell)}\sqrt{2})$ to arrive at (3.46).

Denoting the probability density function of $\sigma^{(k,\ell)}/\sigma^{(\ell,\ell)}$ by p_σ, we get for fixed $\beta > 0$,

$$\begin{aligned}
\mathrm{P}&\Big[\Big|\Big\langle \mathbf{x}_j^{(k)}, \mathbf{x}_i^{(\ell)} \Big\rangle\Big| \geq \Big|\Big\langle \mathbf{x}_v^{(\ell)}, \mathbf{x}_i^{(\ell)} \Big\rangle\Big|\Big] \\
&= \int_0^\infty \left(1 - \mathrm{P}\Big[\Big|\Big\langle \mathbf{x}_j^{(k)}, \mathbf{x}_i^{(\ell)} \Big\rangle\Big| < \Big|\Big\langle \mathbf{x}_v^{(\ell)}, \mathbf{x}_i^{(\ell)} \Big\rangle\Big| \Big| x\Big]\right) p_\sigma(x) \mathrm{d}x \\
&= \int_0^\infty \frac{2}{\pi} \arctan(x)\, p_\sigma(x) \mathrm{d}x \\
&\geq \int_\beta^\infty \frac{2}{\pi} \arctan(x)\, p_\sigma(x) \mathrm{d}x \\
&\geq \frac{2}{\pi} \arctan(\beta) \int_\beta^\infty p_\sigma(x) \mathrm{d}x \\
&= \frac{2}{\pi} \arctan(\beta)\, \mathrm{P}\left[\frac{\sigma^{(k,\ell)}}{\sigma^{(\ell,\ell)}} \geq \beta\right]. \qquad (3.47)
\end{aligned}$$

We continue by setting

$$\beta := \frac{\sqrt{\mathrm{tr}\big(\tilde{\mathbf{R}}^{(k)} \tilde{\mathbf{R}}^{(\ell)}\big)}}{5\sqrt{3}\sqrt{\mathrm{tr}\big(\tilde{\mathbf{R}}^{(\ell)} \tilde{\mathbf{R}}^{(\ell)}\big)}}$$

and obtain

$$\begin{aligned}
\mathrm{P}\left[\frac{\sigma^{(k,\ell)}}{\sigma^{(\ell,\ell)}} \geq \beta\right] &\geq \mathrm{P}\left[\left\{\sigma^{(k,\ell)} \geq \frac{1}{\sqrt{3}} \sqrt{\mathrm{tr}\big(\tilde{\mathbf{R}}^{(k)} \tilde{\mathbf{R}}^{(\ell)}\big)}\right\}\right. \\
&\qquad\qquad \left. \cap \left\{\sigma^{(\ell,\ell)} \leq 5\sqrt{\mathrm{tr}\big(\tilde{\mathbf{R}}^{(\ell)} \tilde{\mathbf{R}}^{(\ell)}\big)}\right\}\right] \\
&\geq 1 - \mathrm{P}\left[\sigma^{(k,\ell)} < \frac{1}{\sqrt{3}} \sqrt{\mathrm{tr}\big(\tilde{\mathbf{R}}^{(k)} \tilde{\mathbf{R}}^{(\ell)}\big)}\right] \\
&\qquad - \mathrm{P}\left[\sigma^{(\ell,\ell)} > 5\sqrt{\mathrm{tr}\big(\tilde{\mathbf{R}}^{(\ell)} \tilde{\mathbf{R}}^{(\ell)}\big)}\right] \\
&> 1 - e^{-\frac{1}{9}} - e^{-8} > \frac{1}{10}, \qquad (3.48)
\end{aligned}$$

where the second inequality follows from a union bound argument, and the third from

$$\mathrm{P}\left[\sigma^{(k,\ell)} < \frac{1}{\sqrt{3}}\sqrt{\operatorname{tr}\big(\tilde{\mathbf{R}}^{(k)}\tilde{\mathbf{R}}^{(\ell)}\big)}\right] \leq e^{-\frac{1}{9}} \tag{3.49}$$

and

$$\mathrm{P}\left[\sigma^{(\ell,\ell)} > 5\sqrt{\operatorname{tr}\big(\tilde{\mathbf{R}}^{(\ell)}\tilde{\mathbf{R}}^{(\ell)}\big)}\right] \leq e^{-8}, \tag{3.50}$$

both proven below. Inserting (3.48) into (3.47) yields the desired result.

Proof of (3.49): We start by noting that ${\sigma^{(k,\ell)}}^2 = \|{\mathbf{C}^{(k)}}^\top \mathbf{C}^{(\ell)}\mathbf{y}_i^{(\ell)}\|_2^2 = {\mathbf{y}_i^{(\ell)}}^\top {\mathbf{C}^{(\ell)}}^\top \tilde{\mathbf{R}}^{(k)}\mathbf{C}^{(\ell)}\mathbf{y}_i^{(\ell)}$ can be written as ${\sigma^{(k,\ell)}}^2 \sim \sum_{m=1}^{M} \lambda_m z_m^2$, where λ_m, $m \in [M]$, denotes the non-negative eigenvalues of ${\mathbf{C}^{(\ell)}}^\top \tilde{\mathbf{R}}^{(k)}\mathbf{C}^{(\ell)}$ and z_m, $m \in [M]$, are independent standard normal random variables. Setting $\boldsymbol{\lambda} = [\lambda_1 \ldots \lambda_M]^\top$ and applying the lower tail bound (Laurent and Massart, 2000, Lem. 1) for linear combinations of independent χ^2 random variables yields, for $t > 0$,

$$\mathrm{P}\left[{\sigma^{(k,\ell)}}^2 \leq \|\boldsymbol{\lambda}\|_1 - 2\|\boldsymbol{\lambda}\|_2\sqrt{t}\right] \leq e^{-t}. \tag{3.51}$$

The inequality (3.49) is obtained from (3.51) by noting that $\|\boldsymbol{\lambda}\|_2 \leq \|\boldsymbol{\lambda}\|_1$ and

$$\|\boldsymbol{\lambda}\|_1 = \operatorname{tr}\big({\mathbf{C}^{(\ell)}}^\top \tilde{\mathbf{R}}^{(k)}\mathbf{C}^{(\ell)}\big) = \operatorname{tr}\big(\tilde{\mathbf{R}}^{(k)}\mathbf{C}^{(\ell)}{\mathbf{C}^{(\ell)}}^\top\big) = \operatorname{tr}\big(\tilde{\mathbf{R}}^{(k)}\tilde{\mathbf{R}}^{(\ell)}\big),$$

and by setting $t = 1/9$ in (3.51).

Proof of (3.50): Noting that $\sigma^{(\ell,\ell)} = f(\mathbf{y}_i^{(\ell)}) = \|{\mathbf{C}^{(\ell)}}^\top \mathbf{C}^{(\ell)}\mathbf{y}_i^{(\ell)}\|_2 = \|\tilde{\mathbf{R}}^{(\ell)}\mathbf{y}_i^{(\ell)}\|_2$ is Lipschitz with Lipschitz constant $\|\tilde{\mathbf{R}}^{(\ell)}\|_{2\to 2}$, we can invoke a well-known concentration inequality for Lipschitz functions of Gaussian random vectors with independent standard normal entries (see, e.g., (Foucart and Rauhut, 2013, Thm. 8.40)) to get, for $t > 0$,

$$\mathrm{P}\left[\left\|\tilde{\mathbf{R}}^{(\ell)}\mathbf{y}_i^{(\ell)}\right\|_2 - \mathbb{E}\left[\left\|\tilde{\mathbf{R}}^{(\ell)}\mathbf{y}_i^{(\ell)}\right\|_2\right] \geq t\right] \leq \exp\left(-\frac{t^2}{2\|\tilde{\mathbf{R}}^{(\ell)}\|_{2\to 2}^2}\right). \tag{3.52}$$

The inequality (3.50) is now implied by

$$\mathbb{E}[\|\tilde{\mathbf{R}}^{(\ell)}\mathbf{y}_i^{(\ell)}\|_2] \leq \sqrt{\mathbb{E}\left[\|\tilde{\mathbf{R}}^{(\ell)}\mathbf{y}_i^{(\ell)}\|_2^2\right]} = \sqrt{\operatorname{tr}(\tilde{\mathbf{R}}^{(\ell)}\tilde{\mathbf{R}}^{(\ell)})} = \|\tilde{\mathbf{R}}^{(\ell)}\|_F$$

(where we used Jensen's inequality), $\|\tilde{\mathbf{R}}^{(\ell)}\|_{2\to 2} \leq \|\tilde{\mathbf{R}}^{(\ell)}\|_F$, and (3.52) with $t = 4\|\tilde{\mathbf{R}}^{(\ell)}\|_F$.

CHAPTER 4

Deep Generative Models for Distribution-Preserving Lossy Compression

We propose and study the problem of distribution-preserving lossy compression. Motivated by recent advances in extreme image compression which allow to maintain artifact-free reconstructions even at very low bitrates, we propose to optimize the rate-distortion tradeoff under the constraint that the reconstructed samples follow the distribution of the training data. Such a compression system recovers both ends of the spectrum: On one hand, at zero bitrate it learns a generative model of the data, and at high enough bitrates it achieves perfect reconstruction. Furthermore, for intermediate bitrates it smoothly interpolates between learning a generative model of the training data and perfectly reconstructing the training samples. We study several methods to approximately solve the proposed optimization problem, including a novel combination of WGAN and WAE, and present an extensive theoretical and empirical characterization of the proposed compression systems.

4.1. PROBLEM FORMULATION

Setup: Consider a random variable $X \in \mathcal{X}$ with distribution P_X. The latter could be modeling, for example, natural images, text documents, or audio signals. In standard lossy compression, the goal is to create a rate-constrained *encoder* $E\colon \mathcal{X} \to \mathcal{W} := \{1, \dots, 2^R\}$, mapping the *input* to a *code* of R bits, and a *decoder* $D\colon \mathcal{W} \to \mathcal{X}$, mapping the code back to the input space, such as to minimize some distortion measure $d\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$. Formally, one aims at solving

$$\min_{E,D} \quad \mathbb{E}_X[d(X, D(E(X)))]. \tag{4.1}$$

In the classic lossy compression setting, both E and D are typically deterministic. As a result, the number of distinct reconstructed inputs $\hat{X} := D(E(X))$ is bounded by 2^R. The main drawback is, as R decreases, the reconstruction $\hat{X}$ will incur increasing degradations (such as blur or blocking in the case of natural images), and will be constant for $R = 0$. Note that simply allowing E, D in (4.1) to be stochastic does not resolve this problem as discussed in Section 4.2.

Distribution-preserving lossy compression: Motivated by recent advances in extreme image compression (Agustsson et al., 2018), we propose and study a novel compression problem: Solve (4.1) under the constraint that the distribution of reconstructed instances $\hat{X}$ follows the distribution of the training data X. Formally, we want to solve the problem

$$\min_{E,D} \quad \mathbb{E}_{X,D}[d(X, D(E(X)))] \quad \text{s.t.} \quad D(E(X)) \sim X, \tag{4.2}$$

where the decoder is allowed to be stochastic.[1] The goal of the distribution matching constraint is to enforce artifact-free reconstructions for all rates. Furthermore, as the rate $R \to 0$, the solution converges to a generative model of X, while for sufficiently large rates R the solution guarantees perfect reconstruction and trivially satisfies the distribution constraint.

[1] Note that a stochastic decoder is necessary if P_X is continuous.

4.2. DEEP GENERATIVE MODELS FOR DISTRIBUTION-PRESERVING LOSSY COMPRESSION

The distribution constraint makes solving the problem (4.2) extremely challenging, as it amounts to learning an exact generative model of the generally unknown distribution P_X for $R = 0$. As a remedy, one can relax the problem and consider the regularized formulation,

$$\min_{E,D} \quad \mathbb{E}_{X,D}[d(X, D(E(X)))] + \lambda d_f(P_{\hat{X}}, P_X), \tag{4.3}$$

where $\hat{X} = D(E(X))$, and d_f is a (statistical) divergence that can be estimated from samples using, e.g., the GAN framework (Goodfellow et al., 2014).

Challenges of the extreme compression regime: At any finite rate R, the distortion term and the divergence term in (4.3) have strikingly opposing effects. In particular, for distortion measures for which $\min_y d(x, y)$ has a unique minimizer for every x, the decoder minimizing the distortion term is constant, conditioned on the code w. For example, if $d(x, y) = \|x - y\|^2$, the optimal decoder D for a fixed encoder E obeys $D(w) = \mathbb{E}_X[X|E(X) = w]$, i.e., it is biased to output the mean. For many popular distortions measures, D, E minimizing the distortion term therefore produce reconstructions $\hat{X}$ that follow a *discrete* distribution, which is at odds with the often *continuous* nature of the data distribution. In contrast, the distribution divergence term encourages $D \circ E$ to generate outputs that are as close as possible to the data distribution P_X, i.e., it encourages $D \circ E$ to follow a continuous distribution if P_X is continuous. While in practice the distortion term can have a stabilizing effect on the optimization of the divergence term (see (Agustsson et al., 2018)), it discourages the decoder form being stochastic—the decoder learns to ignore the noise fed as an input to provide stochasticity, and does so even when adjusting λ to compensate for the increase in distortion when R decreases (see the experiments in Section 4.4). This is in line with recent results for deep generative models in conditional settings: As soon as they are provided with context information, they tend to ignore stochasticity as discussed in (Zhu et al., 2017a; Mathieu et al., 2016), and in particular (Zhu et al., 2017b) and references therein.

Proposed method: We propose and study different generative model-based approaches to approximately solve the DPLC problem. These approaches overcome the aforementioned problems and can be applied for all bitrates R, enabling a gentle tradeoff between matching the distribution of the training data and perfectly reconstructing the training samples. Figure 4.1 provides an overview of the proposed method.

In order to mitigate the bias-to-the-mean-issues with relaxations of the form (4.3), we decompose D as $D = G \circ B$, where G is a generative model taking samples from a fixed prior distribution P_Z as an input, trained to minimize a divergence between $P_{G(Z)}$ and P_X, and B is a stochastic function that is trained together with E to minimize distortion for a fixed G.

Out of the plethora of divergences commonly used for learning generative models G (Kingma and Welling, 2014; Goodfellow et al., 2014), the Wasserstein distance between $P_{\hat{X}}$ and P_X is particularly well suited for DPLC. In fact, it has a distinct advantage as it can be defined for an arbitrary transportation cost function, in particular for the distortion measure d quantifying the quality of the reconstruction in (4.2). For this choice of transportation cost, we can *analytically* quantify the distortion as a function of the rate and the Wasserstein distance between $P_{G(Z)}$ and P_X.

Learning the generative model G: The Wasserstein distance between two distributions P_X and P_Y w.r.t. the measurable cost function $c\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ is defined as

$$W_c(P_X, P_Y) := \inf_{\Pi \in \mathcal{P}(P_X, P_Y)} \mathbb{E}_{(X,Y)\sim\Pi}[c(X, Y)], \tag{4.4}$$

where $\mathcal{P}(P_X, P_Y)$ is a set of all joint distributions of (X, Y) with marginals P_X and P_Y, respectively. When $(\mathcal{X}, d')$ is a metric space and we set $c(x, y) = d'(x, y)$ we have by Kantorovich-Rubinstein duality (Villani, 2008) that

$$W_{d'}(P_X, P_Y) := \sup_{f \in \mathcal{F}_1} \mathbb{E}_X[f(X)] - \mathbb{E}_Y[f(Y)], \tag{4.5}$$

where $\mathcal{F}_1$ is the class of bounded 1-Lipschitz functions $f \colon \mathcal{X} \to \mathbb{R}$. Let $G : \mathcal{Z} \to \mathcal{X}$ and set $Y = G(Z)$ in (4.5), where Z is distributed according to the prior distribution P_Z. Minimizing the latter over the parameters of the mapping G, one recovers the Wasserstein GAN (WGAN) proposed in (Arjovsky et al., 2017). On the other hand, for $Y = G(Z)$ with deterministic G, (4.4) is equivalent to factorizing the couplings $\mathcal{P}(P_X, P_{G(Z)})$ through Z using a conditional distribution function $Q(Z|X)$ (with Z-marginal $Q_Z(Z)$) and minimizing over $Q(Z|X)$ (Tolstikhin et al., 2018), i.e.,

$$\inf_{\Pi \in \mathcal{P}(P_X, P_{G(Z)})} \mathbb{E}_{(X,Y)\sim\Pi}[c(X, Y)] = \inf_{Q\colon\, Q_Z = P_Z} \mathbb{E}_X \mathbb{E}_{Q(Z|X)}[c(X, G(Z))]. \tag{4.6}$$

In this model, the so-called *Wasserstein Autoencoder (WAE)*, $Q(Z|X)$ is parametrized as the push-forward of P_X, through some possibly stochastic function $F \colon \mathcal{X} \to \mathcal{Z}$ and (4.6) becomes

$$\inf_{F\colon\, F(X)\sim P_Z} \mathbb{E}_X \mathbb{E}_F[c(X, G(F(X)))], \tag{4.7}$$

which is then minimized over G.

Note that, in order to solve (4.2), one cannot simply set $c(x, y) = d(x, y)$ and replace F in (4.7) with a rate-constrained version $\hat{F} = B \circ E$, where E is a rate-constrained encoder as introduced in Section 4.1 and $B \colon \mathcal{W} \to \mathcal{Z}$ a stochastic function. Indeed, the tuple $(X, G(F(X)))$ in (4.7) parametrizes the couplings $\mathcal{P}(P_X, P_{G(Z)})$ and $G \circ F$ should therefore be of high model capacity. Using $\hat{F}$ instead of F severely constrains the model capacity of $G \circ \hat{F}$ (for small R) compared to $G \circ F$, and minimizing (4.7) over $G \circ \hat{F}$ would hence not compute a

$G(Z)$ which approximately minimizes $W_c(P_X, P_{G(Z)})$.

Learning the function $B \circ E$: To circumvent this issue, instead of replacing F in (4.7) by $\hat{F}$, we propose to first learn $G^\star$ by either minimizing the primal form (4.6) via WAE or the dual form (4.5) via WGAN (if d is a metric) for $c(x, y) = d(x, y)$, and *subsequently* minimize the distortion as

$$\min_{B,E\colon\, B(E(X)) \sim P_Z} \mathbb{E}_{X,B}[d(X, G^\star(B(E(X))))] \tag{4.8}$$

w.r.t. the fixed generator $G^\star$. We then recover the stochastic decoder D in (4.2) as $D = G^\star \circ B$. Clearly, the distribution constraint in (4.8) ensures that $G^\star(B(E(X))) \sim G^\star(Z)$ since G was trained to map P_Z to P_X.

Reconstructing the Wasserstein distance: The proposed method has the following guarantees.

Theorem 4.1. *Suppose $\mathcal{Z} = \mathbb{R}^m$ and $\|\cdot\|$ is a norm on $\mathbb{R}^m$. Further, assume that $\mathbb{E}[\|Z\|^{1+\delta}] < \infty$ for some $\delta > 0$, let d be a metric and let $G^\star$ be K-Lipschitz, i.e., $d(G^\star(x), G^\star(y)) \leq K\|x - y\|$. Then,*

$$\begin{aligned} W_d(P_X, P_{G^\star(Z)}) &\leq \min_{B,E\colon\, B(E(X)) \sim P_Z} \mathbb{E}_{X,B}[d(X, G^\star(B(E(X))))] \\ &\leq W_d(P_X, P_{G^\star(Z)}) + 2^{-R/m} KC, \end{aligned} \tag{4.9}$$

where $C > 0$ is an absolute constant that depends on $\delta, m, \mathbb{E}[\|Z\|^{1+\delta}]$, and $\|\cdot\|$. Furthermore, for an arbitrary distortion measure d and arbitrary $G^\star$ it holds for all $R \geq 0$

$$W_d(P_X, P_{G^\star(B(E(X)))}) = W_d(P_X, P_{G^\star(Z)}). \tag{4.10}$$

The proof is presented in Appendix 4.A. Theorem 4.1 states that the distortion incurred by the proposed procedure is equal to $W_d(P_X, P_{G^\star(Z)})$ up to an additive error term that decays exponentially in R, hence converging to $W_d(P_X, P_{G^\star(Z)})$ as $R \to \infty$. Intuitively, as E is no longer rate-constrained asymptotically, we can replace F in (4.6) by $B \circ E$ and our two-step procedure is equivalent to minimizing (4.7) w.r.t. G, which amounts to minimizing $W_d(P_X, P_{G(Z)})$ w.r.t. G by (4.6).

Furthermore, according to Theorem 4.1, the distribution mismatch between $G^\star(B(E(X))$ and P_X is determined by the quality of the generative model $G^\star$, and is *independent* of R. This is natural given that we learn $G^\star$ independently.

We note that the proof of (4.9) in Theorem 4.1 hinges upon the fact that W_d is defined w.r.t. the distortion measure d. The bound can also be applied to a generator G' obtained by minimizing, e.g., some f-divergence (Liese and Miescke, 2007) between P_X and $P_{G(Z)}$. However, if $W_d(P_X, P_{G'(Z)}) > W_d(P_X, P_{G^\star(Z)})$ (which will generally be the case in practice) then the distortion obtained by using G' will asymptotically be larger than that obtained for $G^\star$. This suggests using W_d rather than f-divergences to learn G.

4.3. UNSUPERVISED TRAINING VIA WASSERSTEIN++

To learn G, B, and E from data, we parametrize each component as a DNN and solve the corresponding optimization problems via stochastic gradient descent (SGD). We embed the code $\mathcal{W}$ as vectors (henceforth referred to as "centers") in Euclidean space. Note that the centers can also be learned from the data (Agustsson et al., 2017). Here, we simply fix them to the set of vectors $\{-1, 1\}^R$ and use the differentiable approximation from (Mentzer et al., 2018) to backpropagate gradients through this non-differentiable embedding. To ensure that the mapping B is stochastic, we feed noise together with the (embedded) code $E(X)$.

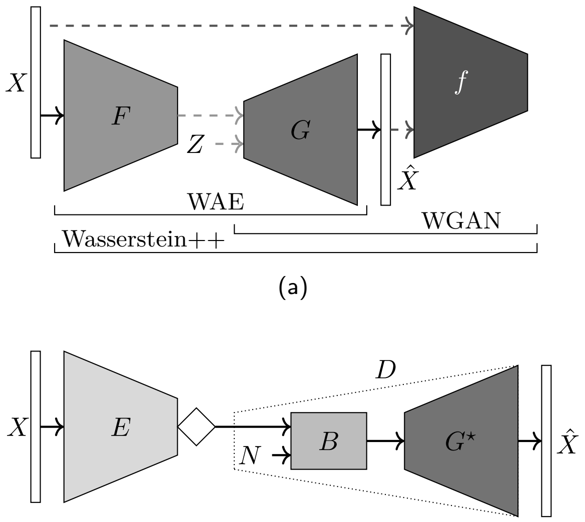
The distribution constraint in (4.8), i.e., ensuring that $B(E(X)) \sim P_Z$, can be implemented using a maximum mean discrepancy (MMD) (Gretton et al., 2012) or GAN-based (Tolstikhin et al., 2018) regularizer. Firstly, we note that both MMD and GAN-based regularizers can be learned from the samples—for MMD via the corresponding U-estimator, and for GAN via the adversarial framework. Secondly, matching the (simple) prior distribution P_Z is much easier than matching the likely complex distribution P_X as in (4.3). Intuitively, at high rates, B should learn to ignore the noise at its input and map the code to P_Z. On the other hand, as $R \to 0$, the code becomes low-dimensional and B is forced to combine it with the stochasticity of the noise at its input to match P_Z. In practice, we observe that MMD is robust and allows to enforce P_Z at all rates R, while GAN-based regularizers are prone to mode collapse at low rates.

Wasserstein++: As previously discussed, $G^\star$ can be learned via WGAN (Arjovsky et al., 2017) or WAE (Tolstikhin et al., 2018). As the WAE framework naturally includes an encoder, it ensures that the structure of the latent space $\mathcal{Z}$ is amenable to encode into. On the other hand, there is no reason that such a structure should emerge in the latent space of G trained via WGAN (in particular when $\mathcal{Z}$ is high-dimensional).[2] In our experiments we observed that WAE tends to produce somewhat less sharp samples than WGAN. On the other hand, WAE is arguably less prone to mode dropping than WGAN as the WAE objective severely penalizes mode dropping due to the reconstruction error term. To combine the best of both approaches, we propose the following novel combination of the primal and the dual form of W_d, via their convex combination

$$\begin{aligned} W_c(P_X, P_G(Z)) = \gamma &\left(\sup_{f \in \mathcal{F}_1} \mathbb{E}_X[f(X)] - \mathbb{E}_Y[f(G(Z))] \right) \\ &+ (1-\gamma) \left(\inf_{F\colon F(X) \sim P_Z} \mathbb{E}_X \mathbb{E}_F[d(X, G(F(X)))] \right), \end{aligned} \tag{4.11}$$

with $\gamma \in [0,1]$. There are two practical questions remaining. Firstly, minimizing this expression w.r.t. G can be done by alternating between performing gradient updates for the critic f and gradient

[2] In principle, this is not an issue if B has enough model capacity, but it might lead to differences in practice as the distortion (4.8) should be easier to minimize if the $\mathcal{Z}$-space is suitably structured, see Section 4.4.



(a)

(b)

Fig. 4.1: (a) A generative model G of the data distribution is commonly learned by minimizing the Wasserstein distance between P_X and $P_{G(Z)}$ either (i) via Wasserstein autoencoder (WAE) (Tolstikhin et al., 2018), where $G \circ F$ parametrizes the couplings between P_X and $P_{G(Z)}$, or (ii) via Wasserstein GAN (WGAN) (Arjovsky et al., 2017), which relies on the critic f. We propose *Wasserstein++*, a novel approach subsuming both WAE and WGAN. (b) Combining the trained generative model $G^\star$ with a rate-constrained encoder E (quantization denoted by $\diamondsuit$-symbol), and a stochastic function B (stochasticity is provided through the noise vector N) to realize a distribution-preserving compression (DPLC) system which minimizes the distortion between X and $\hat{X}$, while ensuring that P_X and $P_{\hat{X}}$ are similar at all rates.

updates for G, F. In other words, we combine the steps of the WGAN algorithm (Arjovsky et al., 2017, Algorithm 1) and WAE-MMD algorithm (Tolstikhin et al., 2018, Algorithm 2), and call this combined algorithm Wasserstein++. Secondly, one can train the critic f on fake samples from $G(Z)$ or from $G(F(X))$, which will not follow the same distribution in general due to a mismatch between $F(X)$ and

P_Z, which is more pronounced in the beginning of the optimization process. Preliminary experiments suggest that the following setup yields samples of best quality (in terms of FID score):

(i) Train f on samples from $G(\tilde{Z})$, where $\tilde{Z} = UZ + (1-U)F(X)$ with $U \sim \text{Uniform}(0,1)$.

(ii) Train G only on samples from $F(X)$, for both the WGAN and the WAE loss term.

We note that training f on samples from $G(\tilde{Z})$ instead of $G(Z)$ arguably introduces robustness to distribution mismatch in $\mathcal{Z}$-space. A more detailed description of Wasserstein++ can be found in Appendix 4.C, and the relation of Wasserstein++ to existing approaches combining GANs and autoencoders is discussed in Section 4.5. We proceed to present the empirical evaluation of the proposed approach.

4.4. EMPIRICAL EVALUATION[3]

Setup: We empirically evaluate the proposed DPLC framework for $G^\star$ trained via WAE-MMD (with an inverse multiquadratics kernel, see (Tolstikhin et al., 2018)), WGAN with gradient penalty (WGAN-GP) (Gulrajani et al., 2017), and Wasserstein++ (implementing the 1-Lipschitz constraint in (4.11) via the gradient penalty from (Gulrajani et al., 2017)), on two standard generative modeling benchmark image datasets, CelebA (Liu et al., 2015b) and LSUN bedrooms (Yu et al., 2015), both downscaled to 64×64 resolution. We focus on these data sets at relatively low resolution as current state-of-the-art generative models can handle them reasonably well, and we do not want to limit ourselves by the difficulties arising with generative models at higher resolutions. The Euclidean distance is used as distortion measure (training objective) d in all experiments.

We measure the quality of the reconstructions of our DPLC systems via mean squared error (MSE) and we assess how well the distribution of the testing reconstructions matches that of the original

[3] Code is available at `https://github.com/mitscha/dplc`.

data using the FID score, which is the recommended measure for image data (Heusel et al., 2017; Lucic et al., 2018). To quantify the variability of the reconstructions conditionally on the code w (i.e., conditionally on the encoder input), we estimate the mean conditional pixel variance $\mathrm{PV}\,[\hat{X}|w] = \frac{1}{N}\sum_{i,j}\mathbb{E}_B[(\hat{X}_{i,j} - \mathbb{E}_B[\hat{X}_{i,j}|w])^2|w]$, where N is the number of pixels of X. In other words, PV is a proxy for how well $G \circ B$ picks up the noise at its input at low rates. All performance measures are computed on a testing set of 10k samples held out form the respective training set, except PV which is computed on a subset 256 testing samples, averaged over 100 reconstructions per testing sample (i.e., code w).

Architectures, hyperparameters, and optimizer: The prior P_Z is an m-dimensional multivariate standard normal, and the noise vector providing stochasticity to B has m i.i.d. entries distributed uniformly on $[0, 1]$. We use the DCGAN (Radford et al., 2015) generator and discriminator architecture for G and f, respectively. For F and E we follow (Tolstikhin et al., 2018) and apply the architecture similar to the DCGAN discriminator. B is realized as a stack of n residual blocks (He et al., 2016a). We set $m = 128$, $n = 2$ for CelebA, and $m = 512$, $n = 4$ for the LSUN bedrooms data set. We chose m to be larger than the standard latent space dimension for GANs as we observed that lower m may lead to blurry reconstructions.

As baselines, we consider compressive autoencoders (CAEs) with the same architecture $G \circ B \circ E$ but without feeding noise to B, training G, B, E jointly to minimize distortion, and BPG (Bellard, 2018), a state-of-the-art engineered codec.[4] In addition, to corroborate the claims made on the disadvantages of (4.3) in Section 4.2, we train $G \circ B \circ E$ to minimize (4.3) as done in the generative compression (GC) approach from (Agustsson et al., 2018), but replacing d_f by W_d.

Throughout, we rely on the Adam optimizer (Kingma and Ba, 2015). To train G by means of WAE-MMD and WGAN-GP we use the

[4] The implementation from (Bellard, 2018) used in this chapter cannot compress to rates below ≈ 0.2 bpp on average for the data sets considered here.

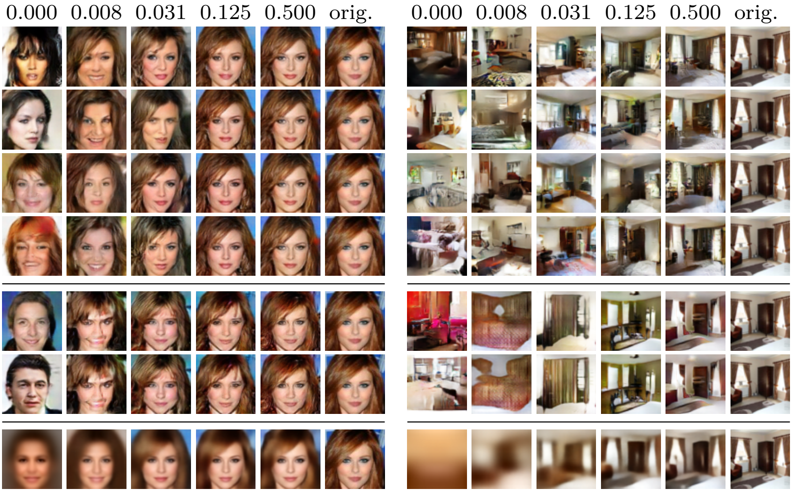


Fig. 4.1: Example (test) reconstructions for CelebA (left) and LSUN bedrooms (right) obtained by our DPLC method based on Wasserstein++ (rows 1–4), and a compressive autoencoder (CAE) baseline (row 5), as a function of the compression rate (in bits per pixel). We stress that even as the bitrate decreases, DPLC manages to generate diverse and realistic-looking images, whereas the CAE reconstructions become increasingly blurry.

training parameters form (Tolstikhin et al., 2018) and (Gulrajani et al., 2017), respectively. For Wasserstein++, we set γ in (4.11) to $2.5 \cdot 10^{-5}$ for CelebA and to 10^{-4} for LSUN. Further, we use the same training parameters to solve (4.8) as for WAE-MMD. Thereby, to compensate for the increase in the reconstruction loss with decreasing rate, we adjust the coefficient of the MMD penalty, λ_{MMD} (see Appendix 4.C), proportionally as a function of the reconstruction loss of the CAE baseline, i.e., $\lambda_{\text{MMD}}(R) = \text{const.} \cdot \text{MSE}_{\text{CAE}}(R)$. We adjust the coefficient λ of the divergence term d_f in (4.3) analogously. This ensures that the regularization strength is roughly the same for all rates. Appendix 4.B provides a detailed description of all architectures and hyperparameters.

Table 4.1: Reconstruction FID and MSE (without rate constraint[5]), and sample FID for the trained generators G, on CelebA and LSUN bedrooms (smaller is better for all three metrics). Wasserstein++ obtains lower rFID and sFID than WAE, but a (slightly) higher sFID than WGAN-GP.

	CelebA			LSUN bedrooms		
	MSE	rFID	sFID	MSE	rFID	sFID
WAE	0.0165	38.55	51.82	0.0099	42.59	153.57
WGAN-GP	/	/	22.70	/	/	45.52
Wasserstein++	**0.0277**	**10.93**	**23.36**	**0.0321**	**27.52**	**60.97**

Results: Table 4.1 shows sample FID of $G^\star$ for WAE, WGAN-GP, and Wasserstein++, as well as the reconstruction FID and MSE for WAE and Wasserstein++.[5] In Figure 4.2 we plot the MSE, the reconstruction FID, and PV obtained by our DPLC models as a function of the bitrate, for different $G^\star$, along with the values obtained for the baselines. Figure 4.1 presents visual examples produced by our DPLC model with $G^\star$ trained using Wasserstein++, along with examples obtained for GC and CAE. More visual examples can be found in Appendix 4.D.

Discussion: We first discuss the performance of the trained generators $G^\star$, shown in Table 4.1. For both CelebA and LSUN bedrooms, the sample FID obtained by Wasserstein++ is considerably smaller than that of WAE, but slightly larger than that of WGAN-GP. Further, Wasserstein++ yields a significantly smaller reconstruction FID than WAE, but a larger reconstruction MSE, as the Wasserstein++ objective is obtained by adding a WGAN term to the WAE objective (which minimizes distortion).

We now turn to the DPLC results obtained for CelebA shown in Figure 4.2, top row. It can be seen that among our DPLC models, the

[5] The reconstruction FID and MSE in Table 4.1 are obtained as $G^\star(F(X))$, without rate constraint. We do not report reconstruction FID and MSE for WGAN-GP as its formulation (4.5) does not naturally include an unconstrained encoder.

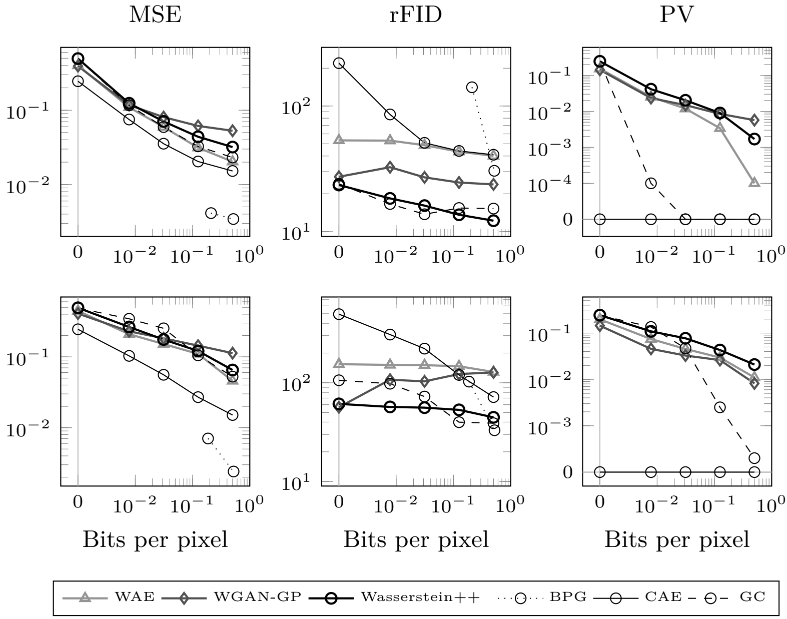


Fig. 4.2: Testing MSE (smaller is better), reconstruction FID (smaller is better), conditional pixel variance (PV, larger is better) obtained by our DPLC model, for different generators $G^\star$, CAE, BPG, as well as GC (Agustsson et al., 2018), as function of the bitrate. The results for CelebA are shown in the top row, those for LSUN bedrooms in the bottom row. The PV of our DPLC models steadily increases with decreasing rate, i.e., they generate gradually more image content, as opposed to GC.

one combined with $G^\star$ from WAE yields the lowest MSE, followed by those based on Wasserstein++, and WGAN-GP. This is not surprising as the optimization of WGAN-GP does not include a distortion term. CAE obtains a lower MSE than all DPLC models which is again intuitive as G, B, E are trained jointly and to minimize distortion exclusively (in particular there is no constraint on the distribution in $\mathscr{Z}$-space). Finally, BPG obtains the overall lowest MSE. Note, however, that BPG relies on several advanced techniques such as entropy coding based on context models (see, e.g., (Rippel and Bourdev, 2017; Li et al., 2018; Mentzer et al., 2018; Ballé et al., 2018)), which we did not implement here (but which could be incorporated into our DPLC framework).

Among our DPLC methods, DPLC based on Wasserstein++ attains the lowest reconstruction FID (i.e., its distribution most faithfully reproduces the data distribution) followed by WGAN-GP and WAE. For all three models, the FID decreases as the rate increases, meaning that the models manage *not only to reduce distortion as the rate increases, but also to better reproduce the original distribution.* The FID of CAE increases drastically as the rate falls below 0.03 bpp. Arguably, this can be attributed to significant blur incurred at these low rates (see Figure 4.D.9 in Appendix 4.D). BPG yields a very high FID as soon as the rate falls below 0.5 bpp due to compression artifacts.

The PV can be seen to increase steadily for all DPLC models as the rate decreases, as expected. This is also reflected by the visual examples in Figure 4.1, left: At 0.5 bpp no variability is visible, at 0.125 bpp the facial expression starts to vary, and decreasing the rate further leads to the encoder producing different persons, deviating more and more form the original image, until the system generates random faces.

In contrast, the PV obtained by solving (4.3) as in GC (Agustsson et al., 2018) is essentially 0, except at 0 bpp, where it is comparable to that of our DPLC models. The noise injected into $D = G \circ B$ is hence ignored unless it is the only source of randomness at 0 bpp. We emphasize that this is the case even though we adjust the coefficient λ of the d_f term as $\lambda(R) = \text{const.} \cdot \text{MSE}_{\text{CAE}}(R)$ to compensate for the increase in distortion with decreasing rate. The performance of GC in terms of MSE and reconstruction FID is comparable to that of the DPLC model with Wasserstein++ $G^\star$.

We now turn to the DPLC results obtained for LSUN bedrooms. The qualitative behavior of DPLC based on WAE and Wasserstein++ in terms of MSE, reconstruction FID, and PV is essentially the same as observed for CelebA. Wasserstein++ provides the lowest FID by a large margin, for all positive rates. The reconstruction FID for WAE is high at all rates, which is not surprising as the sample FID obtained by WAE is large (cf. Table 4.1), i.e., WAE struggles to model the distribution of the LSUN bedrooms data set.

For DPLC based on WGAN-GP, in contrast, while the MSE and PV follow the same trend as for CelebA, the reconstruction FID increases notably as the bitrate decreases. By inspecting the corresponding reconstructions (cf. Figure 4.D.12 in Appendix 4.D) one can see that the model manages to approximate the data distribution well at zero bitrate, but yields increasingly blurry reconstructions as the bitrate increases. This indicates that either the (trained) function $B \circ E$ is not mapping the original images to $\mathcal{Z}$ space in a way suitable for $G^\star$ to produce crisp reconstructions, or the range of $G^\star$ does not cover the support of P_X well. We tried to address the former issue by increasing the depth of B (to increase model capacity) and by increasing λ_{MMD} (to reduce the mismatch between the distribution of $B(E(X))$ and P_Z), but we did not observe improvements in reconstruction quality. We therefore suspect mode coverage issues to cause the blur in the reconstructions.

Finally, GC (Agustsson et al., 2018) largely ignores the noise injected into D at high bitrates, while using it to produce stochastic decoders at low bitrates. However, at low rates, the rFID of GC is considerably higher than that of DPLC based on Wasserstein++, meaning that it does not faithfully reproduce the data distribution despite using stochasticity. Indeed, GC suffers from mode collapse at low rates as can be seen in Figure 4.D.14 in Appendix 4.D.

4.5. RELATED WORK

DNN-based methods for compression have become an active area of research over the past few years. Most authors focus on image compression (Toderici et al., 2015, 2017; Theis et al., 2017; Rippel and Bourdev, 2017; Ballé et al., 2017; Agustsson et al., 2017; Li et al., 2018; Johnston et al., 2017; Torfason et al., 2018; Mentzer et al., 2018; Ballé et al., 2018), while others consider audio (Kankanahalli, 2018) and video (Wu et al., 2018) data. Compressive autoencoders (Theis et al., 2017; Ballé et al., 2017; Agustsson et al., 2017; Li et al., 2018; Torfason et al., 2018; Ballé et al., 2018) and recurrent neural networks (RNNs) (Toderici et al., 2015, 2017; Johnston et al., 2017) have emerged as the most popular DNN architectures for compression.

GANs have been used in the context of learned image compression before (Rippel and Bourdev, 2017; Agustsson et al., 2018; Santurkar et al., 2018; Galteri et al., 2017). Rippel and Bourdev (2017) apply a GAN loss to image patches for artifact suppression, whereas Agustsson et al. (2018) apply the GAN loss to the entire image to encourage the decoder to generate image content (but does not demonstrate a properly working stochastic decoder). GANs are leveraged by Ledig et al. (2017) and Galteri et al. (2017) to improve image quality of super resolution and engineered compression methods, respectively.

Santurkar et al. (2018) use a generator trained with a GAN as a decoder in a compression system. However, they rely on vanilla GAN (Goodfellow et al., 2014) only rather than considering different W_d-based generative models and they do not provide an analytical characterization of their model. Most importantly, they optimize their model using conventional distortion minimization with deterministic decoder, rather than solving the DPLC problem.

Gregor et al. (2016) propose a variational autoencoder (VAE)-type generative model that learns a hierarchy of progressively more abstract representations. By storing the high-level part of the representation and generating the low-level one, they manage to partially preserve and partially generate image content. However, their framework is lacking a notion of rate and distortion and does not quantize the representations into a code (apart from using finite precision data types).

Probably most closely related to Wasserstein++ is VAE-GAN (Larsen et al., 2015), combining the VAE (Kingma and Welling, 2014) with vanilla GAN (Goodfellow et al., 2014). However, whereas the VAE part and the GAN part minimize different divergences (Kullback-Leibler and Jensen-Shannon in the case of VAE and vanilla GAN, respectively), WAE and WGAN minimize the same cost function, so Wasserstein++ is somewhat more principled conceptually. More generally, learning generative models jointly with an inference mechanism for the latent variables has attracted significant attention, see, e.g., (Larsen et al., 2015; Donahue et al., 2017; Dumoulin et al., 2017; Dosovitskiy and Brox, 2016) and (Rosca et al., 2017) for an overview.

Outside of the domain of machine learning, the problem of distribution-preserving (scalar) quantization was studied. Specifically, (Delp and Mitchell, 1991) studies moment preserving quantization, that is quantization with the design criterion that certain moments of the data distribution shall be preserved. Further, (Li et al., 2010) proposes an engineered dither-based quantization method that preserves the distribution of the variable to be quantized.

4.6. CONCLUSION AND FUTURE WORK

In this chapter, we studied the DPLC problem, which amounts to optimizing the rate-distortion tradeoff under the constraint that the reconstructed samples follow the distribution of the training data. We proposed different approaches to solve the DPLC problem, in particular Wasserstein++, a novel combination of WAE and WGAN, and analytically characterized the properties of the resulting compression systems. These systems allowed us to obtain essentially artifact-free reconstructions at all rates, covering the full spectrum from learning a generative model of the data at zero bitrate on one hand, to learning a compression system with almost perfect reconstruction at high bitrate on the other hand. Most importantly, our framework improves over previous methods by producing stochastic decoders at low bitrates, thereby effectively solving the DPLC problem for the first time. Future work includes scaling the proposed approach up to full-resolution images and applying it to data types other than images.

APPENDICES

4.A. PROOF OF THEOREM 4.1

We first prove (4.9). We start by constructing a rate-constrained stochastic function $\hat{F}\colon \mathcal{X} \to \mathbb{R}^m$ as follows. Let q be the nearest neighbor quantizer

$$q(z) = \arg\min_{i \in [2^R]} \|z - c_i\| \tag{4.12}$$

with the centers $\{c_1, \dots, c_{2^R}\} \subset R^m$ chosen to minimize $\mathbb{E}[\min_{i\in[2^R]} \|z - c_i\|]$ (the minimum is attained by Prop. 2.1 in (Luschgy and Pagès, 2002)). Let A_i be the Voronoi region associated with c_i, i.e., $A_i = \{z \in \mathbb{R}^m \colon \|z - c_i\| = \min_{i\in[2^R]} \|z - c_i\|\}$. We now set $E(X) = q(F^\star(X))$ and $B(i) = Z_i$, where $F^\star$ is a minimizer of (4.7) for $G = G^\star$ and $Z_i \sim P_{Z|Z\in A_i}$, independent of Z given A_i. It holds $\hat{F}(X) := B(E(X)) \sim Z$ by construction and the described choice of B, E is feasible for (4.8). We continue by upper-bounding $d(X, G^\star(\hat{F}(X)))$

$$\begin{aligned} d(X, G^\star(\hat{F}(X))) &\leq d(X, G^\star(F^\star(X))) + d(G^\star(F^\star(X)), G^\star(\hat{F}(X))) \\ &\leq d(X, G^\star(F^\star(X))) + K\|F^\star(X) - \hat{F}(X)\| \\ &\leq d(X, G^\star(F^\star(X))) + K\|F^\star(X) - c_{E(X)}\| \\ &\quad + K\|c_{E(X)} - \hat{F}(X)\|. \end{aligned} \tag{4.13}$$

By Cor. 6.7 from (Graf and Luschgy, 2007) we have

$$\mathbb{E}_X[\|F^\star(X) - c_{E(X)}\|] \leq 2^{-R/m}(C_1 \mathbb{E}[\|Z\|^{1+\delta}] + C_2), \tag{4.14}$$

where $\delta > 0$, $2^R > C_3$, and $C_1, C_2, C_3 > 0$ are numerical constants depending on δ, m and $\|\cdot\|$. The same upper bound holds for $\mathbb{E}[\|c_{E(X)} - \hat{F}(X)\|]$ as $E(X) = q(F^\star(X)) = q(\hat{F}(X))$, i.e., $\|c_{E(X)} - \hat{F}(X)\| = \|c_{q(\hat{F}(X))} - \hat{F}(X)\|$, and $\hat{F}(X) \sim Z \sim F^\star(X)$. Taking the expectation on both sides of (4.13) and using (4.14) in the resulting expression yields the upper bound in (4.9). The lower bound is obtained by noting that the minimization (4.7) includes the rate constrained mappings $B \circ E$.

Eq. (4.10) directly follows from the distribution constraint in (4.8), $B(E(X)) \sim P_Z$: This constraint implies $G^\star(B(E(X))) \sim G^\star(Z)$ which in turn implies (4.10).

Remark 4.1. *The construction of the discrete encoder* $\hat{F}$ *in the proof of Theorem 4.1 requires optimal vector quantization, which is generally NP hard. However, if we make stronger assumptions on* P_Z *than done in the statement of Theorem 4.1 one can prove exponential convergence without optimal vector quantization. For example, if* Z *is uniformly distributed on* $[0,1]^m$*,* $R = km$ *for a positive integer* k*, and* $\|\cdot\|$ *is the Euclidean norm, then one can partition* $[0,1]^m$ *into* 2^R *hypercubes of equal edge length* $1/2^k = 1/2^{\frac{R}{m}}$ *and associate the centers* c_i *with the centers of these hypercubes. In this case, the two last terms in* (4.13) *are upper-bounded by* $\sqrt{m} \cdot 2^{-\frac{R}{m}}$*. The quantization error thus converges to* 0 *as* $2^{-\frac{R}{m}}$ *for* $R \to \infty$*.*

4.B. HYPERPARAMETERS AND ARCHITECTURES

Learning the generative model $G^\star$*:* The training parameters used train $G^\star$ using WAE, WGAN-GP, and Wasserstein++ are shown in Table 4.B.1. The parameters for WAE correspond to those used for the WAE-MMD experiments on CelebA in (Tolstikhin et al. (2018), see Appendix C.2), with the difference that we use a batch size of 256 and a slightly modified schedule (note that 41k iterations with a batch size of 256 correspond to roughly 55 epochs with batch size 100, which is suggested by (Tolstikhin et al., 2018)). This does not notably impact the performance WAE (we obtain a slightly lower sample FID than reported in (Tolstikhin et al., 2018, Table 1)). The parameters for WGAN-GP correspond to those recommended for LSUN bedrooms in (Gulrajani et al., 2017, Appendix E).

Learning the function $B \circ E$: The training parameters to solve (4.8) can be found in Table 4.B.2. To solve (4.1) (i.e., to train the CAE baseline) we use the same parameters as for WAE (except that $\lambda_{\text{MMD}} = 0$ as there is no distribution constraint in (4.1)), see Table 4.B.1.

Training the baseline (4.3) *as in* (Agustsson et al., 2018)*:* To solve (4.3) we use the parameters and schedule specified in Table 4.B.1 for Wasserstein++ (except that we do not need λ_{MMD}), and we determine λ in (4.3) based on the bitrate as $\lambda(R) = 2.5 \cdot 10^{-5} \cdot \frac{\text{MSE}_{\text{CAE}}(R)}{\text{MSE}_{\text{CAE}}(0.5\text{bpp})}$ for CelebA and $\lambda(R) = 7.5 \cdot 10^{-5} \cdot \frac{\text{MSE}_{\text{CAE}}(R)}{\text{MSE}_{\text{CAE}}(1\text{bpp})}$ for LSUN bedrooms.

Architectures: We use the following notation. `cxsy-z` stands for a 2D convolution with an `x` × `x` kernel, stride `y`, and `z` filters, followed by the ReLU non-linearity (the ReLU non-linearity is omitted when the convolution is followed by quantization or the tanh non-linearity). The suffixes `b` and `l`, i.e., `cxsyb-z` and `cxsyl-z`, indicate that batch normalization is employed before the non-linearity and layer normalization as well as leaky ReLU with a negative slope of 0.2 instead of ReLU, respectively. `txsyb-z` stands for a transposed 2D convolution with an `x` × `x` kernel, stride `1/y`, and `z` filters, followed by batch normalization and ReLU non-linearity. `fc-z` denotes flattening followed by a fully-connected layer with `z` neurons. `r-z` designates a residual block with `z` filters in each layer. The abbreviations `bn`, `tanh`, and `-q` are used for batch normalization, the tanh non-linearity, and quantization with differentiable approximation for gradient backpropagation, respectively. k is the number of channels of the quantized feature representation (i.e., k determines the bitrate), and the suffix `+n` denotes concatenation of an m-dimensional noise vector with i.i.d. entries uniformly distributed on $[0, 1]$, reshaped as to match the spatial dimension of the feature maps in the corresponding network layer.

- F: `c4s2-64, c4s2b-128, c4s2b-256, c4s2b-512, fc-`m`, bn`

- G: t4s2b-512, t4s2b-256, t4s2b-128, t4s2b-64, t4s2b-64, c3s1-3, tanh

- f: c3s1-64, c4s2l-64, c4s2l-128, c4s2l-256, c4s2l-512, fc-1

- E: c4s2-64, c4s2b-128, c4s2b-256, c4s2b-512, c3s1-k-q+n

- B: c3s1-512, r-512, ..., r-512, fc-m, bn

Table 4.B.1: Adam learning rates α_F, α_G, α_f for the WAE encoder F, the generator G, and the WGAN critic f, respectively, Adam parameters β_1, β_2, MMD regularization coefficient λ_{MMD}, mini-batch size b, number of (generator) iterations n_{iter}, and learning rate schedule (LR sched.), for CelebA. The number of critic iterations per generator iteration is set to $n_{\mathsf{critic}} = 5$ for WGAN-GP and Wasserstein++. **For LSUN bedrooms**, the parameters are identical, except that $\lambda_{\mathsf{MMD}} = 300$ for WAE and Wasserstein++, and the number of iterations is doubled (with the learning rate schedule scaled accordingly) for all three algorithms.

	α_F	α_G	α_f	β_1	β_2	λ_{MMD}	λ_{GP}	b	n_{iter}	LR sched.
WAE	10^{-3}	10^{-3}	/	0.5	0.999	100	/	256	41k	×0.4@22k;38k
WGAN-GP	/	10^{-4}	10^{-4}	0.5	0.900	/	10	64	100k	/
Wasserstein++	$3 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	10^{-4}	0.5	0.999	100	10	256	25k	×0.4@15k;21k

Table 4.B.2: Adam parameters α, β_1, β_2, MMD regularization coefficient λ_{MMD} as a function of the MSE incurred by CAE at rate R ($\mathsf{MSE}_{\mathsf{CAE}}(R)$), mini-batch size b, number of iterations n_{iter}, and learning rate schedule (LR sched.) used to solve (4.8).

	α	β_1	β_2	$\lambda_{\text{MMD}}(R)$	b	n_{iter}	LR sched.
CelebA	10^{-3}	0.5	0.999	$150 \cdot \frac{\text{MSE}_{\text{CAE}}(R)}{\text{MSE}_{\text{CAE}}(0.5\text{bpp})}$	256	41k	×0.4@22k;38k
LSUN bedrooms	10^{-3}	0.5	0.999	$800 \cdot \frac{\text{MSE}_{\text{CAE}}(R)}{\text{MSE}_{\text{CAE}}(1\text{bpp})}$	256	82k	×0.4@44k;76k

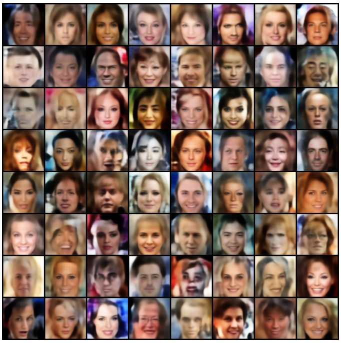
4.C. THE WASSERSTEIN++ ALGORITHM

Algorithm 1: Wasserstein++

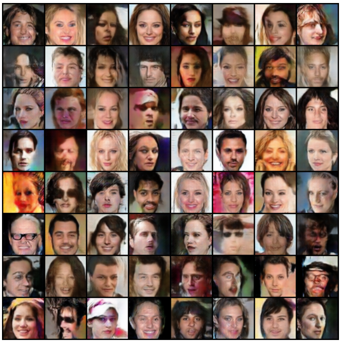
```
Require: MMD regularization coefficient $\lambda_{\mathrm{MMD}}$, WGAN coefficient $\gamma$,
    WGAN gradient penalty coefficient $\lambda_{\mathrm{GP}}$, number of critic iterations
    per generator iteration $n_{\mathrm{critic}}$, mini-batch size $b$, characteristic
    positive-definite kernel $k$, Adam parameters (not shown explicitly).
Initialize the parameters $\phi$, $\theta$, and $\psi$ of the WAE encoder $F_\phi$, the
    generator $G_\theta$, and the WGAN discriminator $f_\psi$, respectively.
while $(\phi, \theta, \psi)$ not converged do
    for $t = 1, \ldots, n_{\mathrm{critic}}$ do
        Sample $\{x_1, \ldots, x_b\}$ from the training set
        Sample $\bar{z}_i$ from $F_\phi(x_i)$, for $i = 1, \ldots, b$
        Sample $\{z_1, \ldots, z_b\}$ from the prior $P_Z$
        Sample $\{\eta_1, \ldots, \eta_b\}$ from Uniform$(0, 1)$
        Sample $\{\nu_1, \ldots, \nu_b\}$ from Uniform$(0, 1)$
        $\tilde{z}_i \leftarrow \eta_i z_i + (1 - \eta_i)\bar{z}_i, \quad \text{for } i = 1, \ldots, b$
        $\hat{x}_i \leftarrow G_\theta(\tilde{z}_i), \quad \text{for } i = 1, \ldots, b$
        $\tilde{x}_i \leftarrow \nu_i x_i + (1 - \nu_i)\hat{x}_i, \quad \text{for } i = 1, \ldots, b$
        $L_f \leftarrow \frac{1}{b} \sum_{i=1}^{b} f_\psi(\hat{x}_i) - f_\psi(x_i) + \lambda_{\mathrm{GP}}(\|\nabla_{\tilde{x}_i} f_\psi(\tilde{x}_i)\| - 1)^2$
        $\psi \leftarrow \mathrm{Adam}(\psi, L_f)$
    end for
    $L_d \leftarrow \frac{1}{b} \sum_{i=1}^{b} \|x_i - G_\theta(\bar{z}_i)\|$
    $L_{\mathrm{MMD}} \leftarrow$
        $\frac{1}{b(b-1)} \sum_{\ell \neq j} k(z_\ell, z_j) + \frac{1}{b(b-1)} \sum_{\ell \neq j} k(\bar{z}_\ell, \bar{z}_j) - \frac{2}{b^2} \sum_{\ell, j} k(z_\ell, \bar{z}_j)$
    $L_{\mathrm{WGAN}} \leftarrow \frac{1}{b} \sum_{i=1}^{b} -f_\psi(G_\theta(\bar{z}_i))$
    $\theta \leftarrow \mathrm{Adam}(\theta, (1 - \gamma)L_d + \gamma L_{\mathrm{WGAN}})$
    $\phi \leftarrow \mathrm{Adam}(\phi, (1 - \gamma)(L_d + \lambda_{\mathrm{MMD}} L_{\mathrm{MMD}}))$
end while
```

4.D. VISUAL EXAMPLES

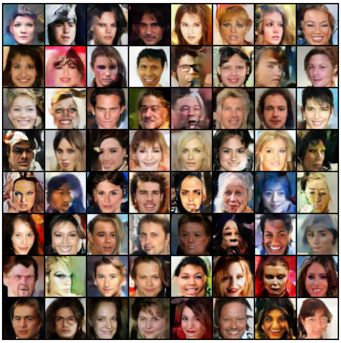
In the following, we show random samples and reconstructions produced by different DPLC models and CAE, at different bitrates, for the CelebA and LSUN bedrooms data set. None of the examples are cherry-picked.



WAE



WGAN-GP



Wasserstein++

Figure 4.D.4: Random samples produced by the trained generator $G^{\star}(Z)$, with $Z \sim P_Z$, on CelebA. The samples produced by WGAN-GP and Wasserstein++ are sharper than those generated by WAE.

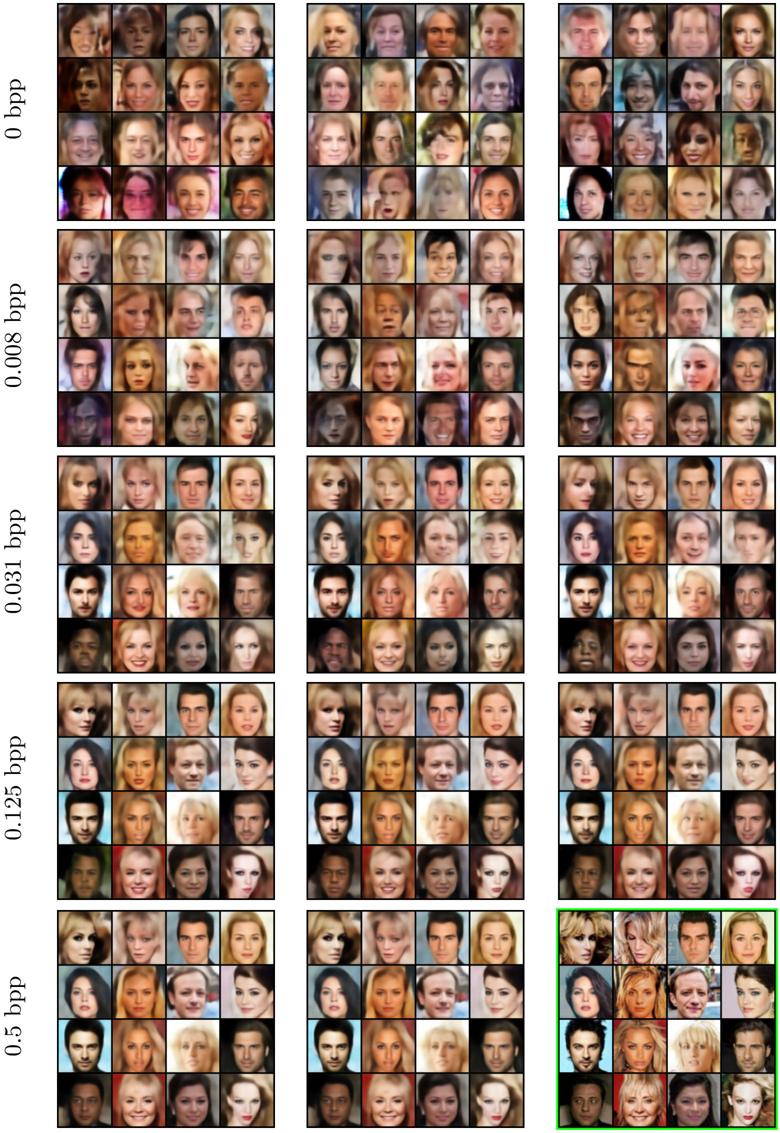


Figure 4.D.5: Testing reconstructions produced by our DPLC model with WAE $G^{\star}$, along with the original image (green border), for CelebA. The variability between different reconstructions increases as the bitrate decreases.

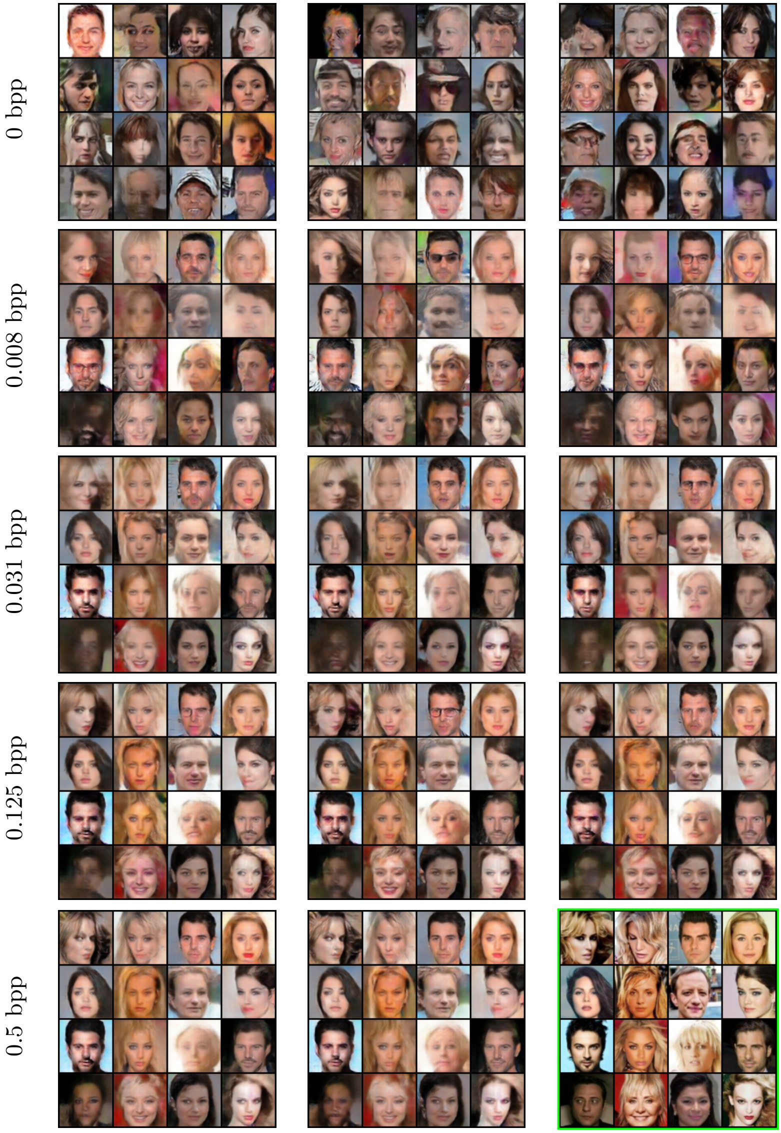


Figure 4.D.6: Testing reconstructions produced by our DPLC model with WGAN-GP $G^{\star}$, along with the original image (green border), for CelebA. The variability between different reconstructions increases as the bitrate decreases.

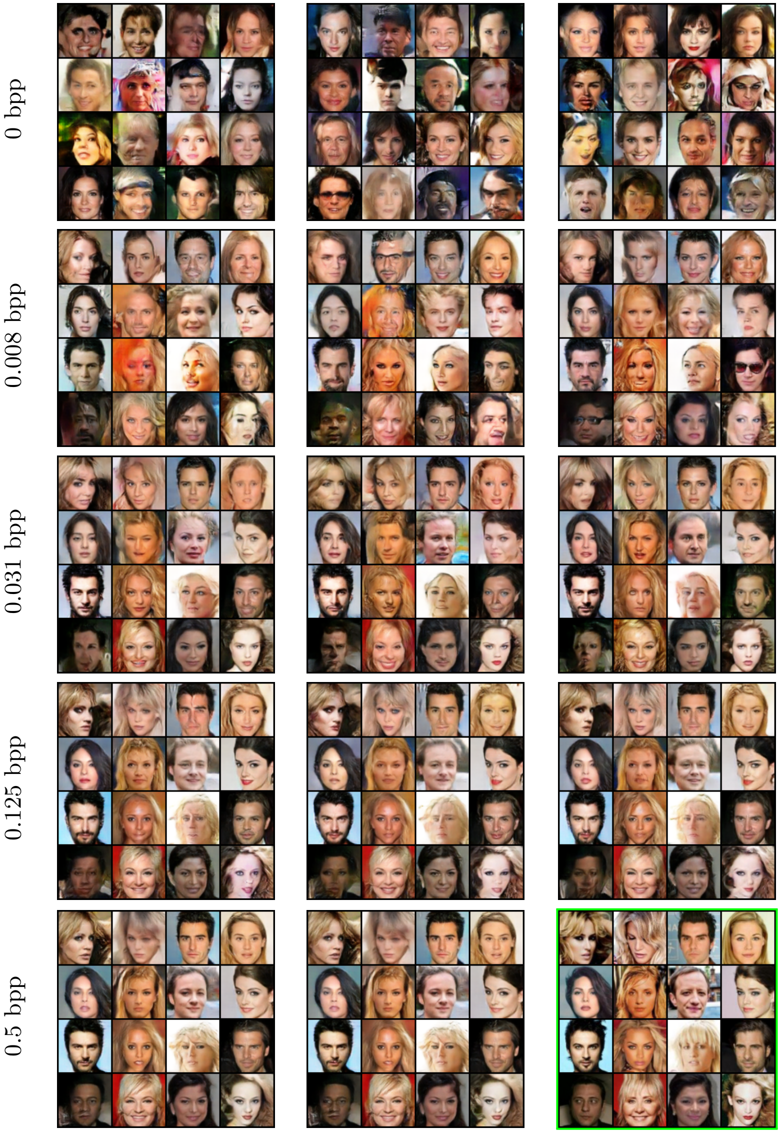


Figure 4.D.7: Testing reconstructions produced by our DPLC model with Wasserstein++ $G^{\star}$, along with the original image (green border), for CelebA. The variability between different reconstructions increases as the bitrate decreases.

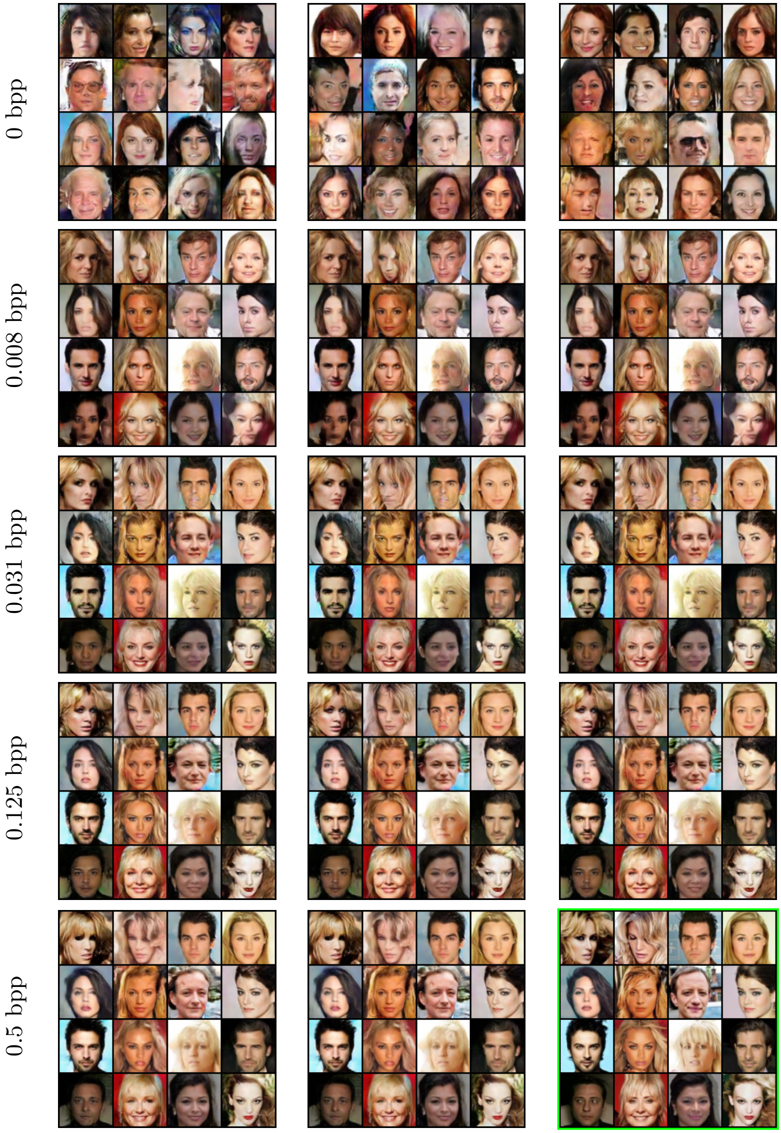
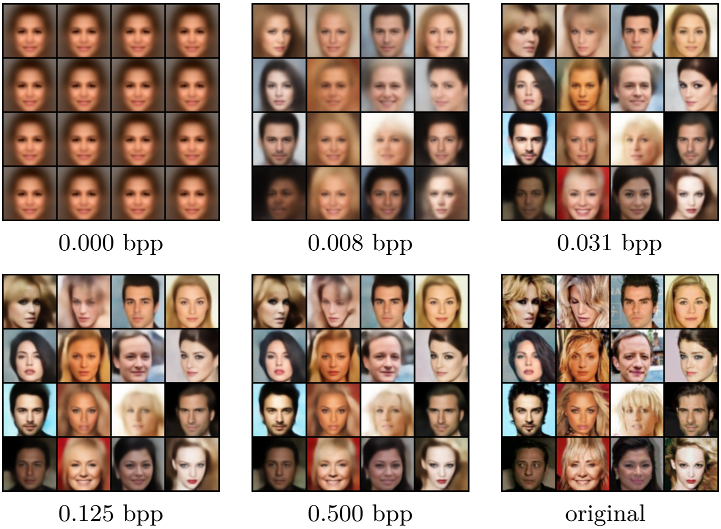


Figure 4.D.8: Testing reconstructions produced using $G \circ B \circ E$ obtained by solving (4.3) similarly as in GC (Agustsson et al., 2018), along with the original image (green border), for CelebA. There is no variability between different reconstructions except at 0 bpp.



0.000 bpp

0.008 bpp

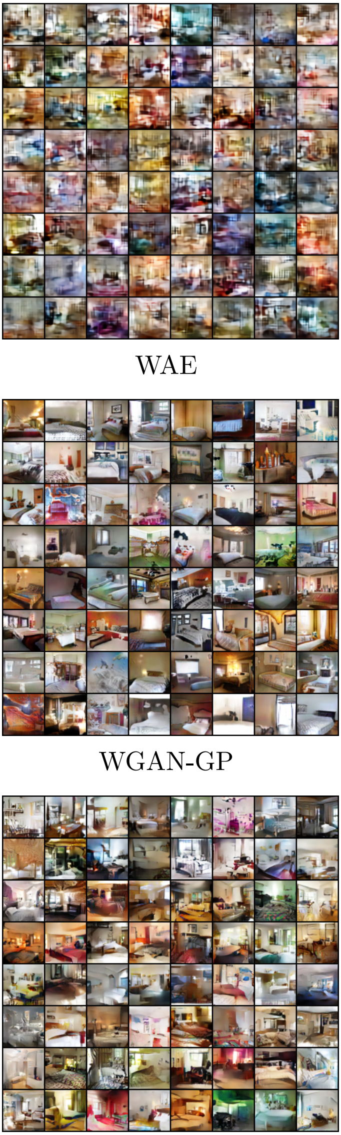
0.031 bpp

0.125 bpp

0.500 bpp

original

Figure 4.D.9: Testing reconstructions produced by CAE, for CelebA. The reconstructions become increasingly blurry as the rate decreases.



WAE

WGAN-GP

Wasserstein++

Figure 4.D.10: Random samples produced by the trained generator $G^{\star}(Z)$, with $Z \sim P_Z$, for LSUN bedrooms. The samples produced by WGAN-GP and Wasserstein++ are sharper than those generated by WAE.

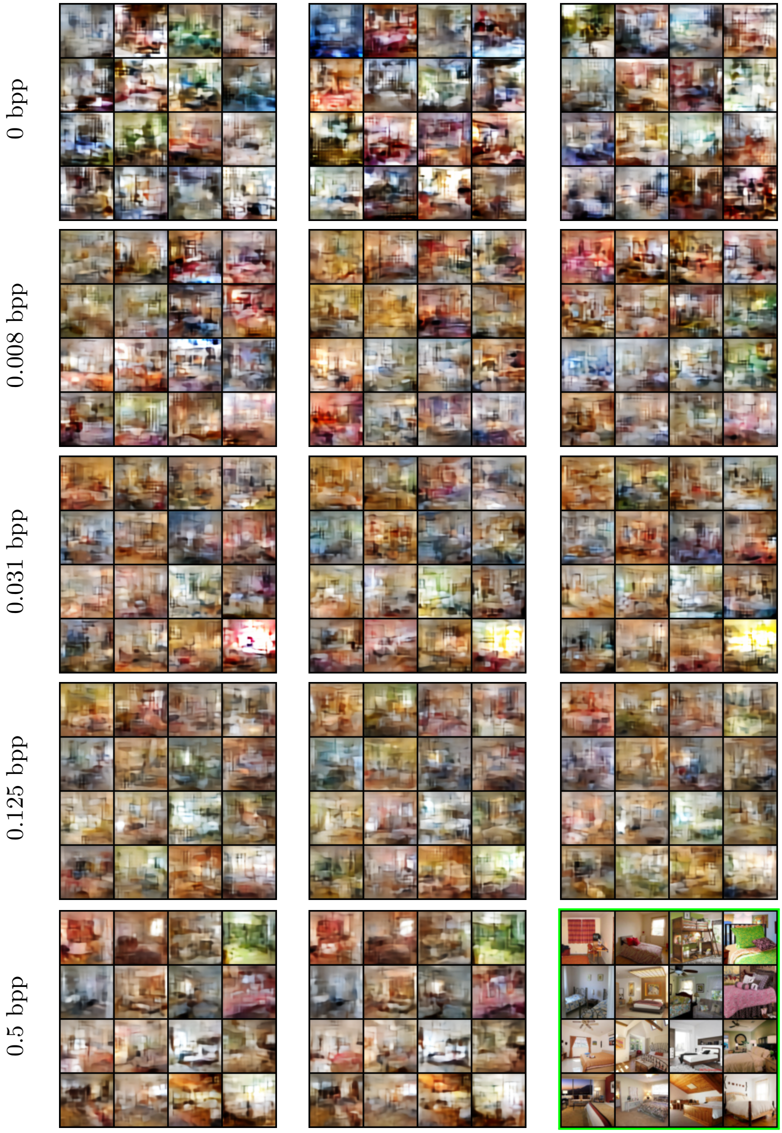


Figure 4.D.11: Testing reconstructions produced by our DPLC model with WGAN-GP $G^{\star}$, along with the original image (green border), for LSUN bedrooms. The reconstructions are blurry at all rates.

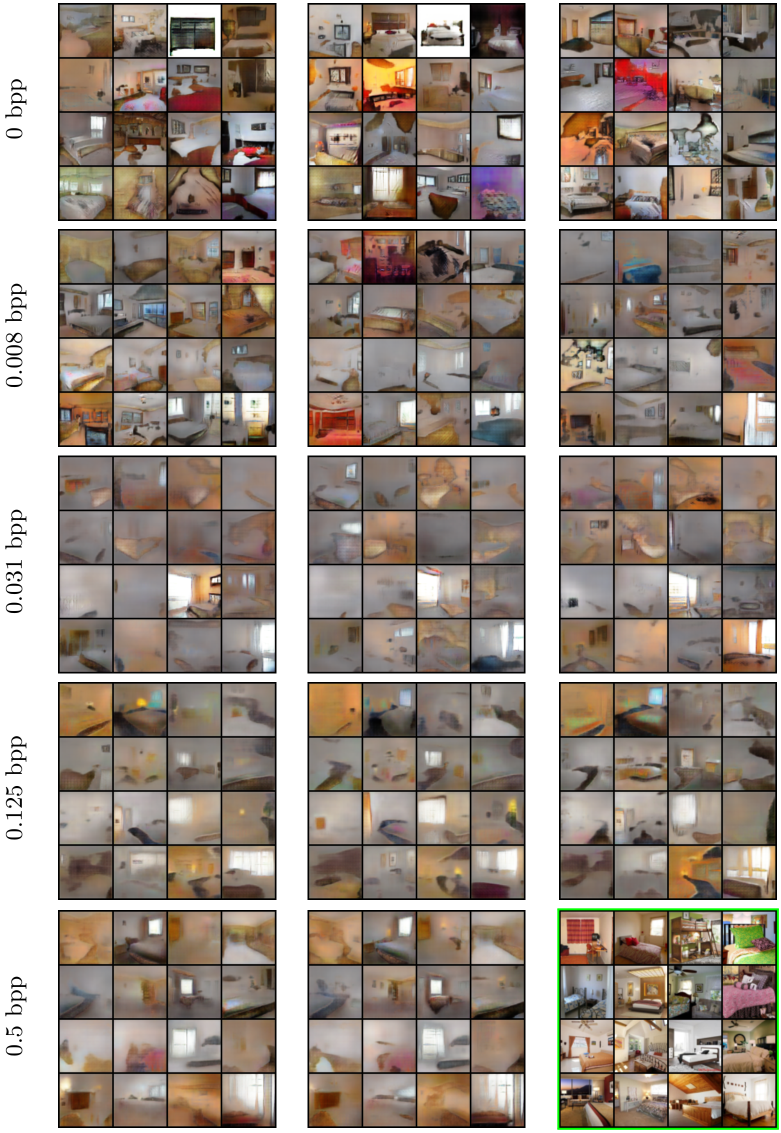


Figure 4.D.12: Testing reconstructions produced by our DPLC model with WGAN-GP $G^{\star}$, along with the original image (green border), for LSUN bedrooms. The reconstructions are blurry at all rates except at 0 bpp.

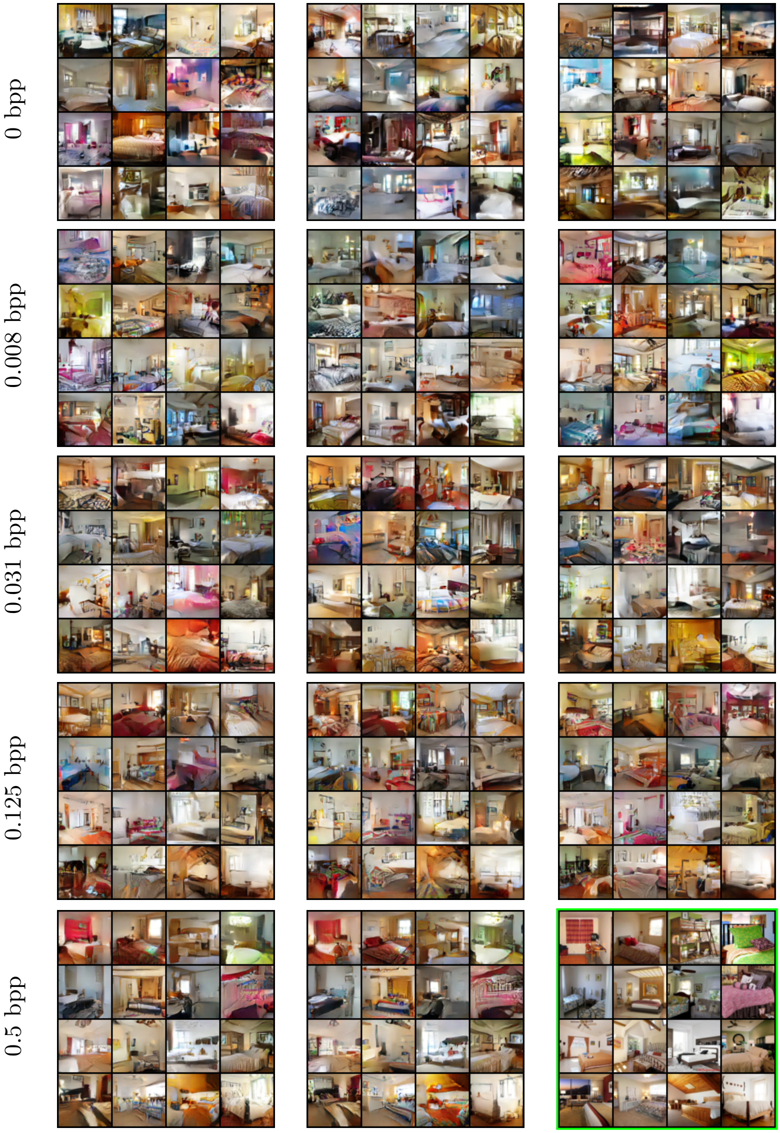


Figure 4.D.13: Testing reconstructions produced by our DPLC model with Wasserstein++ $G^{\star}$, along with the original image (green border), for LSUN bedrooms. The variability between different reconstructions increases as the bitrate decreases. The reconstructions are quite sharp at all rates.

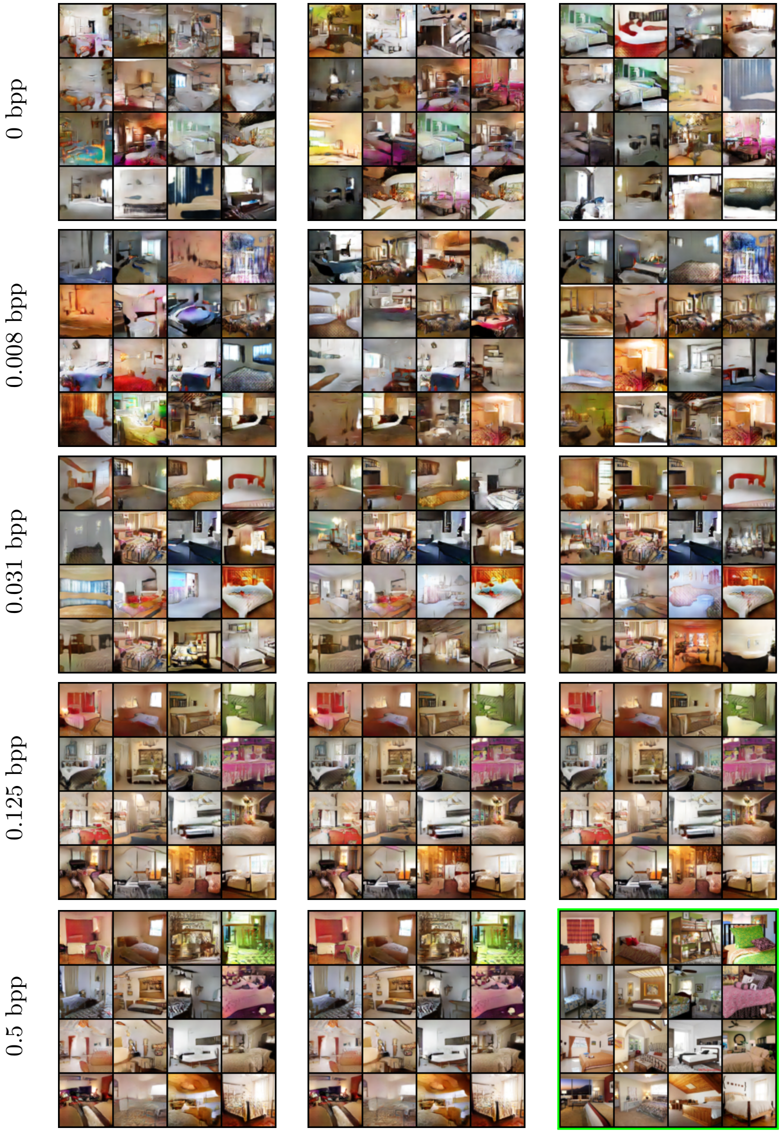
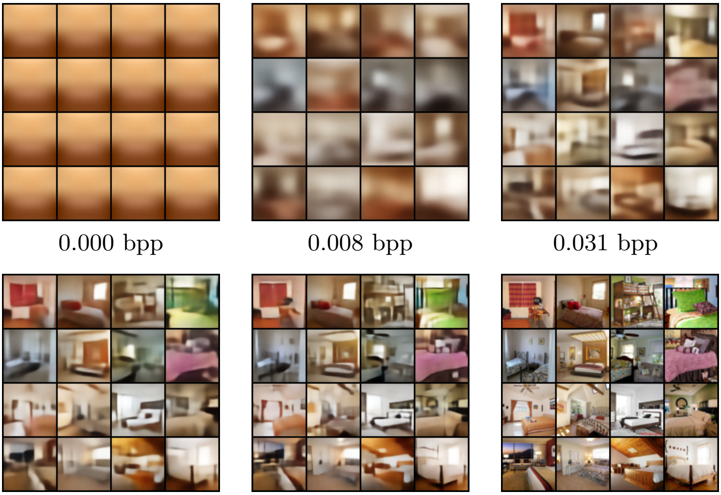


Figure 4.D.14: Testing reconstructions produced using $G \circ B \circ E$ obtained by solving (4.3) similarly as in GC (Agustsson et al., 2018), along with the original image (green border), for LSUN bedrooms. The method produces a stochastic decoder at very low rates but suffers from mode collapse.



0.000 bpp

0.008 bpp

0.031 bpp

0.125 bpp

0.500 bpp

original

Figure 4.D.15: Testing reconstructions produced by CAE, for LSUN bedrooms. The reconstructions become increasingly blurry as the rate decreases.

CHAPTER 5

Deep Learning with a Multiplication Budget

A large fraction of the arithmetic operations required to evaluate DNNs consists of matrix multiplications, in both convolution and fully connected layers. We perform end-to-end learning of low-cost approximations of matrix multiplications in DNN layers by casting matrix multiplications as 2-layer SPNs (arithmetic circuits) and learning their (ternary) edge weights from data. The SPNs disentangle multiplication and addition operations and enable us to impose a budget on the number of multiplication operations. Combining our method with knowledge distillation and applying it to image classification DNNs (trained on ImageNet) and language modeling DNNs (using LSTMs), we obtain a first-of-a-kind reduction in number of multiplications (over 99.5%) while maintaining the predictive performance of the full-precision models. Finally, we demonstrate that the proposed framework is able to rediscover Strassen's matrix multiplication algorithm, learning to multiply 2×2 matrices using only 7 multiplications instead of 8.

5.1. RELATED WORK

We briefly review the most common approaches to compress DNNs, focusing on methods decreasing computational complexity rather than memory footprint. In all cases, there is a tradeoff between the complexity reduction and reduction in the (inference) accuracy of the compressed model.

A popular way to speed up DNNs, in particular convolutional neural networks (CNNs), is to utilize resource-efficient architectures, such as SqueezeNet (Iandola et al., 2016), MobileNet (Howard et al., 2017), and ShuffleNet (Zhang et al., 2017). SqueezeNet reduces the convolution kernel size. MobileNet and ShuffleNet rely on depth-wise separable convolutions and grouped convolutions, respectively. More sophisticated grouping and sharing techniques are studied by Wang et al. (2016a).

Another strategy to accelerate CNNs is to exploit the low-rank structure prevalent in weight matrices and convolution kernels. Denton et al. (2014); Novikov et al. (2015); Kim et al. (2016b) use tensor decompositions to obtain low-rank approximations of pretrained weight matrices and filter tensors, then finetune the approximated weight matrices and filters to restore the accuracy of the compressed models. Other works (Tai et al., 2016; Wen et al., 2017) employ low rank-promoting regularizers to further reduce the rank of the filter tensors. A framework to exploit low-rank structure in the filter responses is presented by Zhang et al. (2016).

Sparsifying filters and pruning channels are popular methods to make DNNs more efficient during inference. Wen et al. (2016) and Lebedev and Lempitsky (2016) rely on group norm-based regularizers and demonstrate their effectiveness in penalizing unimportant filters and channels, promoting hardware-friendly filter shapes, regularizing the network depth, and optimizing the filter receptive fields. Inter-channel and intra-channel redundancy is exploited by Liu et al. (2015a) via a two-stage factorization procedure. An energy-aware methodology to prune filters of CNNs is described in (Yang et al., 2017).

Finally, an effective way to adapt DNNs to resource-constrained platforms is to reduce the numerical precision of their weights and/or activations. Examples for DNNs that quantize both weights and activations are DoReFa-Net (Zhou et al., 2016), XNOR-Net (Rastegari et al., 2016), and ABC-Net (Lin et al., 2017). Other works use binary weights (Courbariaux et al., 2015; Rastegari et al., 2016; Lin et al.,

2017) and ternary weights (Li et al., 2016; Zhu et al., 2016) but maintain full-precision values for the activations. Keeping the activations in full precision instead of quantizing them leads to a smaller decrease in computational cost, but can yield better predictive performance.

5.2. LEARNING FAST MATRIX MULTIPLICATIONS VIA SPNS

5.2.1. Casting matrix multiplication as SPN

Given square matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$, the product $\mathbf{C} = \mathbf{AB}$ can be represented as a 2-layer SPN

$$\operatorname{vec}(\mathbf{C}) = \mathbf{W}_c[(\mathbf{W}_b \operatorname{vec}(\mathbf{B})) \odot (\mathbf{W}_a \operatorname{vec}(\mathbf{A}))] \tag{5.1}$$

where $\mathbf{W}_a, \mathbf{W}_b \in \mathbb{K}^{r \times n^2}$ and $\mathbf{W}_c \in \mathbb{K}^{n^2 \times r}$, with $\mathbb{K} := \{-1, 0, 1\}$, are fixed. The SPN (5.1) disentangles additions (and subtractions), encoded in the ternary matrices $\mathbf{W}_a$, $\mathbf{W}_b$, and $\mathbf{W}_c$, and multiplications, realized exclusively by the operation $\odot$ (see Figure 5.1, left). The width of the hidden layer of the SPN, r, hence determines the number of multiplications used for the matrix multiplication. A naïve implementation of the matrix multiplication $\mathbf{AB}$ requires $r = n^3$. For $n = 2$,[1] Strassen's matrix multiplication algorithm (Strassen, 1969) specifies the following set of weights that satisfy (5.1) for $r = 7$

[1] The formulation by Strassen (1969) is more general, applying recursively to 4 equally-sized subblocks of square matrices, with the 2×2 case occurring at maximal recursion depth.

(instead of $r = 8$)

$$\mathbf{W}_a = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix}, \quad \mathbf{W}_b = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix},$$

$$\mathbf{W}_c = \begin{pmatrix} 1 & 0 & 0 & 1 & -1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}. \tag{5.2}$$

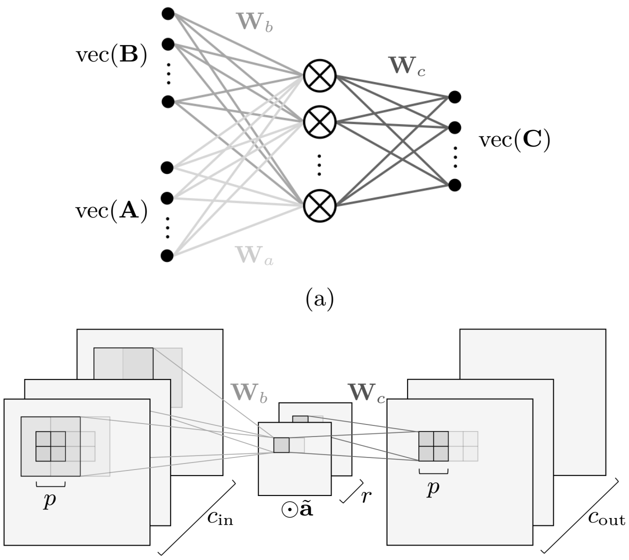
An interesting tensor perspective on the SPN (5.1) (not explored in-depth here) is common in the context of algebraic complexity theory. Specifically, (5.1) can be written as

$$\operatorname{vec}(\mathbf{C})_i = \sum_{k=1}^{n^2} \sum_{\ell=1}^{n^2} (\mathbf{M}_n)_{i,k,\ell} [\operatorname{vec}(\mathbf{A})]_k [\operatorname{vec}(\mathbf{B})]_\ell, \qquad \text{where}$$

$$(\mathbf{M}_n)_{i,k,\ell} = \sum_{j=1}^{r} (\mathbf{W}_c)_{i,j} (\mathbf{W}_a)_{j,k} (\mathbf{W}_b)_{j,\ell}.$$

$\mathbf{M}_n$ is the $(n \times n)$-*matrix multiplication tensor*, and r hence corresponds to the rank of $\mathbf{M}_n$. It is known that $\operatorname{rank}(\mathbf{M}_2) = 7$ and $19 \leq \operatorname{rank}(\mathbf{M}_3) \leq 23$, see (Elser, 2016) for more details and references.

Elser (2016) explores learning exact matrix multiplications via SPNs of the form (5.1) for $n = 2$ and $n = 3$ from synthetic data. Thereby, the elements of $\mathbf{W}_a$, $\mathbf{W}_b$, and $\mathbf{W}_c$ are relaxed to real numbers instead of elements from $\mathbb{K}$. Note that this relaxation leads to an increase in the number of multiplications in general. In contrast, we integrate SPNs with weights from $\mathbb{K}$ into DNN layers and learn them end-to-end (see next section), realizing actual reductions in multiplications.



(a)

(b)

Fig. 5.1: (a) Illustration of the 2-layer SPN (5.1), implementing an (approximate) matrix multiplication. The edges (i.e., the matrices $\mathbf{W}_a$, $\mathbf{W}_b$, $\mathbf{W}_c$) have weights in $\mathbb{K} = \{-1, 0, 1\}$. (b) Application of the proposed framework to 2D convolution leads to p-strided 2D convolution with $\mathbf{W}_b$, followed by channel-wise scaling by $\tilde{\mathbf{a}} = \mathbf{W}_a \text{vec}(\mathbf{A})$, followed by $1/p$-strided transposed 2D convolution with $\mathbf{W}_c$.

5.2.2. Learning fast approximate matrix multiplications for DNNs

Writing matrix products in the form (5.1) is not specific to square matrices. Indeed, it is easy to see that $r \geq nmk$ is a sufficient condition for the existence of matrices $\mathbf{W}_a, \mathbf{W}_b, \mathbf{W}_c$ with elements in $\mathbb{K}$ such that the product of any two matrices $\mathbf{A} \in \mathbb{R}^{k \times m}$ and $\mathbf{B} \in \mathbb{R}^{m \times n}$, including matrix-vector products (i.e., $n = 1$), can be written in the form (5.1). When the matrices $\mathbf{A}$ and $\mathbf{B}$ are drawn from probability distributions that concentrate on low-dimensional manifolds of $\mathbb{R}^{k \times m}$

and $\mathbb{R}^{m\times n}$, respectively, or if one of the matrices is fixed, it may be possible to find $\mathbf{W}_a$ and $\mathbf{W}_b$ that satisfy the equality in (5.1) approximately even when $r \ll nmk$. In this case, (5.1) approximately computes the product $\mathbf{AB}$ while considerably reducing the number of multiplications compared to the naïve implementation. Furthermore, by imposing structure (such as, e.g., sparsity or block-diagonal structure) into the matrices $\mathbf{W}_a$, $\mathbf{W}_b$, $\mathbf{W}_c$ one can tailor sharing or grouping of the operations for the application or platform at hand.

In this chapter, we leverage this concept to accelerate and compress the matrix multiplications in DNN layers for inference. Specifically, for layer ℓ, we associate $\mathbf{A}$ with the (pretrained) weights/filters $\mathbf{W}^\ell$ and $\mathbf{B}$ with the corresponding activations/feature maps $\mathbf{F}^\ell$. The ternary matrices $\mathbf{W}_a$, $\mathbf{W}_b$, and $\mathbf{W}_c$ are then learned end-to-end using a stochastic gradient-based optimizer (one set of weights $\mathbf{W}_a$, $\mathbf{W}_b$, $\mathbf{W}_c$ for each layer). After training, $\mathbf{W}_a$ and $\text{vec}(\mathbf{A})$ can be collapsed into a vector $\tilde{\mathbf{a}} = \mathbf{W}_a\text{vec}(\mathbf{A}) \in \mathbb{R}^r$ as they are both fixed during inference. Alternatively, $\tilde{\mathbf{a}} \in \mathbb{R}^r$, $\mathbf{W}_b$, and $\mathbf{W}_c$ can be learned jointly from scratch. The choice of r determines the tradeoff between the computational cost in terms of multiplications and the precision of the the approximate matrix multiplication, and hence the predictive performance of the network. This approach requires r full-precision parameters and $rm(k+n)$ ternary weight parameters. It reduces the number of multiplications by a factor of mnk/r.

Quantizing the elements of $\mathbf{W}_a$, $\mathbf{W}_b$, and $\mathbf{W}_c$ to $\mathbb{K}$ during training poses a challenge as quantization is non-differentiable. Different approaches were proposed to overcome this issue (Courbariaux et al., 2015; Li et al., 2016; Rastegari et al., 2016; Zhu et al., 2016; Agustsson et al., 2017). Here, we adopt the method from (Li et al., 2016) and briefly describe it for quantizing $\mathbf{W}_a$ ($\mathbf{W}_b$ and $\mathbf{W}_c$ are quantized in exactly the same way). Specifically, this method maintains a full-precision version $\mathbf{W}_a^{\text{fp}}$ of $\mathbf{W}_a$ during training and quantizes $\mathbf{W}_a^{\text{fp}}$ in every forward pass by approximately solving the optimization problem

$$\alpha^*, \mathbf{W}_a^{\text{t}*} = \underset{\alpha, \mathbf{W}_a^{\text{t}}}{\arg\min} \|\mathbf{W}_a^{\text{fp}} - \alpha \mathbf{W}_a^{\text{t}}\|_F^2$$

$$\text{s.t.} \quad \alpha > 0, \quad \mathbf{W}_a^{\mathrm{t}} \in \mathbb{K}^{r \times km}, \tag{5.3}$$

and by setting $\mathbf{W}_a = \alpha^* \mathbf{W}_a^{\mathrm{t}*}$ (the scaling factors α^* for $\mathbf{W}_a$, $\mathbf{W}_b$, $\mathbf{W}_c$ can be absorbed by $\mathbf{A}$ or $\tilde{\mathbf{a}}$ after training to ensure that $\mathbf{W}_a$, $\mathbf{W}_b$, $\mathbf{W}_c$ have elements in $\mathbb{K}$). During the backward pass the quantization function is replaced by the identity function, and the gradient step is applied to $\mathbf{W}_a^{\mathrm{fp}}$. Assuming i.i.d. Gaussian weights, Li et al. (2016) derive the approximate solution

$$(\mathbf{W}_a^{\mathrm{t}*})_{i,j} = \begin{cases} 1 & \text{if } (\mathbf{W}_a^{\mathrm{fp}})_{i,j} > \Delta, \\ -1 & \text{if } (\mathbf{W}_a^{\mathrm{fp}})_{i,j} < -\Delta, \\ 0 & \text{otherwise,} \end{cases}$$
$$\alpha^* = \frac{\sum_{(i,j)\colon (\mathbf{W}_a^{\mathrm{t}*})_{i,j} \neq 0} |(\mathbf{W}_a^{\mathrm{fp}})_{i,j}|}{\sum_{i,j} |(\mathbf{W}_a^{\mathrm{t}*})_{i,j}|} \tag{5.4}$$

to (5.3), where $\Delta = \frac{0.7}{kmr} \sum_{i,j} |(\mathbf{W}_a^{\mathrm{fp}})_{i,j}|$. While our framework would allow quantized training from scratch with fixed threshold Δ and fixed quantization level α (e.g., $\Delta = 0.5$ and $\alpha = 1$), we observed that relying on the scheme (5.4) allows us to pretrain $\mathbf{W}_a^{\mathrm{fp}}$, $\mathbf{W}_b^{\mathrm{fp}}$, $\mathbf{W}_c^{\mathrm{fp}}$ without quantization, and then activate quantization to stably continue training. We found that this strategy leads to faster training while inducing no loss in accuracy.

Besides the fully connected case described in this section, we particularize the proposed approach for 2D convolutions for image classification DNNs. We emphasize that any DNN layer operation reducible to a general matrix multiplication (GEMM) can be cast into the form (5.1), including n-dimensional convolutions, group (equivariant) convolutions (when implemented as a filter bank) (Cohen and Welling, 2016), and deformable convolutions (Dai et al., 2017).

5.2.3. Knowledge distillation (KD)

KD refers to the process of training a student network using a larger (in terms of the number of layers and hidden units) teacher network

(Bucilua et al., 2006; Hinton et al., 2014). As a result, the student network typically has the same or slightly better predictive performance than the teacher network, despite being less complex. KD for training a low-precision student network from a full-precision teacher network with the same architecture and hyper parameters as the student network was investigated recently in (Mishra and Marr, 2018; Zhuang et al., 2018; Polino et al., 2018). Here, we explore the same avenue to improve the predictive performance of networks compressed with our method. Specifically, we follow the method proposed in (Hinton et al., 2014) using the cross entropy between the student softmax output and the teacher softmax output as KD loss term. We set the softmax temperature parameter to 1 throughout and assign the same weight to the KD loss term as to the original loss. For sequence models, we simply apply the described KD loss to the softmax outputs of the unrolled teacher and student models (more sophisticated techniques were proposed in (Kim and Rush, 2016)).

5.2.4. Application to 2D convolution

Consider the ℓth 2D convolution layer of a CNN applying c_{out} filters of dimension $w \times h \times c_{\text{in}}$ to a feature representation $\mathbf{F}^\ell$ of dimension $W \times H \times c_{\text{in}}$ (width×height×number of channels). To write the computation of all c_{out} output channels as a matrix multiplication, each feature map in $\mathbf{F}^\ell$ is decomposed into WH patches of size $w \times h$ (after appropriate padding) and the vectorized patches are arranged in a matrix $\tilde{\mathbf{F}}^\ell$ of dimension $whc_{\text{in}} \times WH$. This transformation is usually referred to as `im2col`, see (Sze et al. (2017), Figure 19) for an illustration. Accordingly, the filters for all output channels are vectorized and jointly reshaped into a $c_{\text{out}} \times whc_{\text{in}}$ matrix $\tilde{\mathbf{W}}^\ell$. The vectorized layer output (before activation) for all c_{out} output channels is obtained as $\tilde{\mathbf{W}}^\ell \tilde{\mathbf{F}}^\ell$ and has dimension $c_{\text{out}} \times WH$. In principle, one can now compress the operation $\tilde{\mathbf{W}}^\ell \tilde{\mathbf{F}}^\ell$ using our method by setting $\mathbf{A} = \tilde{\mathbf{W}}^\ell$, $\mathbf{B} = \tilde{\mathbf{F}}^\ell$, plugging them into (5.1), and proceeding as described in Section 5.2.2. However, this results in impractically large $\mathbf{W}_a$, $\mathbf{W}_b$, and $\mathbf{W}_c$ and ignores the weight sharing structure of the convolution. By associating $\mathbf{A}$ with $\tilde{\mathbf{W}}^\ell$ and $\mathbf{B}$ with single columns of $\tilde{\mathbf{F}}^\ell$ we can jointly compress the computations across all input and output channels, while preserving the spatial structure of the convolution. The resulting SPN realizes a convolution with r ternary $w \times h \times c_{\text{in}}$ filters (the rows of $\mathbf{W}_b$), followed by a channel-wise scaling with $\tilde{\mathbf{a}} = \mathbf{W}_a \text{vec}(\tilde{\mathbf{W}}^\ell)$, followed by convolution with a ternary $1 \times 1 \times r$ filter for each of the c_{out} outputs (the rows of $\mathbf{W}_c$) see Figure 5.1, right.

To realize *local spatial compression*, we partition the computation of the convolution into subsets corresponding to square output patches. In more detail, we consider the computation of $p \times p$ convolution output patches from $(p-1+w) \times (p-1+h)$ input patches, offset by a stride of p, and approximate this computation with a SPN jointly for all channels. As a result, the number of multiplications is reduced both spatially and across channels. For example, for 3×3 convolution filters, we divide the input feature maps into 4×4 spatial patches with a stride of 2, such that the SPN computes $2 \times 2 \times c_{\text{out}}$ outputs from $4 \times 4 \times c_{\text{in}}$ elements of $\mathbf{F}^\ell$. Thereby, $\mathbf{W}_c$ realizes a $2 \times 2 \times r$ transposed convolution with a stride of $1/2$ (see Figure 5.1, right, and pseudocode in Appendix 5.C). For fixed r, this reduces the number of multiplications by a factor of 4 compared to the case without spatial compression (i.e., $p = 1$).

In summary, the described compression of 2D convolution leads to a reduction of the number of multiplications by a factor $c_{\text{in}} c_{\text{out}} whp^2/r$ compared to the standard implementation of the convolution.

Finally, to reduce the number of additions realized through $\mathbf{W}_b$ (and thereby the number of nonzero elements of $\mathbf{W}_b$) by a factor of g, we implement $\mathbf{W}_b$ as grouped convolution, originally introduced in (Krizhevsky et al., 2012). Specifically, the convolution realized by $\mathbf{W}_b$ is assumed to consist of g independent 2D convolutions each with c_{in}/g input channels and r/g output channels. In other words, $\mathbf{W}_b$ is assumed to be block-diagonal with blocks of dimension $(r/g) \times (whc_{\text{in}}/g)$.

Relation to prior work in the 2D convolution case: Binary weight networks (BWNs) (Rastegari et al., 2016) and ternary weight networks (TWNs) (Li et al., 2016) rely on binary $\{-1, 1\}$ and ternary $\{-1, 0, 1\}$ weight matrices, respectively, followed by (full-precision) rescaling of the activations (see Section 5.2.2) and are special cases of our framework. ABC-Nets (Lin et al., 2017) approximate the full-precision weight matrices as a weighted sum of multiple binary $\{-1, 1\}$ weight matrices and can also be cast as (structured) SPNs. However, we do not directly recover the trained ternary quantization (TTQ) approach from (Zhu et al., 2016), which relies on asymmetric ternary weights $\{-c_1, 0, c_2\}$, $c_1, c_2 > 0$. Finally, note that Winograd filter-based convolution (Lavin and Gray, 2016) realizes spatial compression over 2×2 output patches but performs exact computation and does not compress across channels.

5.3. EXPERIMENTS[2]

5.3.1. Rediscovering Strassen's algorithm

Before applying the proposed method to DNNs, we demonstrate that it is able to rediscover Strassen's algorithm, i.e., it can learn to multiply 2×2 matrices using only 7 multiplications instead of 8 (which implies a recursive algorithm for larger matrices). This problem was previously studied by Elser (2016), but for *real-valued* $\mathbf{W}_a$, $\mathbf{W}_b$, $\mathbf{W}_c$, which increases the number of multiplications in general when using these matrices in (5.1) to compute matrix products. In contrast, our method learns $\mathbf{W}_a, \mathbf{W}_b \in \mathbb{K}^{7\times 4}$, $\mathbf{W}_c \in \mathbb{K}^{4\times 7}$ (i.e., the discrete solution space has size $3^{3\cdot 4\cdot 7} = 3^{84}$), and hence leads to an actual reduction in the number of multiplications.

We generate a training set containing 100k pairs $(\mathbf{A}_i, \mathbf{B}_i)$ with entries i.i.d. uniform on $[-1, 1]$, train the SPN with full-precision weights (initialized i.i.d. uniform on $[-1, 1]$) for one epoch with SGD (learning

[2] Code to reproduce the experiments is available at `https://github.com/mitscha/strassennets`.

rate 0.1, momentum 0.9, mini-batch size 4), activate quantization, and train for another epoch (with learning rate 0.001). Around 25 random initializations are necessary to obtain convergence to zero training L2-loss after activation of the quantization; for most initializations the training L2-loss converges to a positive value. A set of ternary weight matrices implementing an *exact* matrix multiplication, found by our method, is

$$\mathbf{W}_a = \begin{pmatrix} -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & 0 & 1 & 0 \\ -1 & -1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}, \quad \mathbf{W}_b = \begin{pmatrix} -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ -1 & -1 & -1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix},$$

$$\mathbf{W}_c = \begin{pmatrix} 1 & 0 & 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & -1 \\ -1 & 0 & 0 & 0 & 1 & 1 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

5.3.2. Image classification

We apply our method to all convolution layers (including the first convolution layer and the projection layers for subsampling) of the ResNet architecture (He et al., 2016a) to create the so-called Strassen-ResNet (ST-ResNet). We evaluate ST-ResNet on CIFAR-10 (10 classes, 50k training images, 10k testing images) (Krizhevsky and Hinton, 2009) and ImageNet (ILSVRC2012; 1k classes, 1.2M training images, 50k testing images) (Russakovsky et al., 2015) for different choices of r, p, g, and compare the accuracy of ST-ResNet to related works. All models were trained from scratch, meaning we directly learn $\tilde{\mathbf{a}} = \mathbf{W}_a \text{vec}(\mathbf{A})$ rather than associating $\mathbf{A}$ with the weights of pretrained networks and learning $\mathbf{W}_a$. Throughout the training process we used SGD with momentum 0.9 and weight decay 10^{-4}. As most related works involving ternary weights do not report sparsity levels, to facilitate comparisons, we do not make any assumption about the number of zeros among ternary weights. It is the sparsity of the activations, not the weights, that directly impacts the number of multiplications (the focus of this chapter). All model sizes are computed without (lossless) compression of the network parameters.

CIFAR-10

We consider ST-ResNet-20 and employ the data augmentation procedure described in (He et al. (2016a), Sec. 4.2.). We train for 250 epochs with initial learning rate 0.1 and mini-batch size 128, multiplying the learning rate by 0.1 after 150 and 200 epochs. We then activate quantization for $\mathbf{W}_b$ and $\mathbf{W}_c$, set the learning rate to 0.01 and train the network for 40 epochs, multiplying the learning rate by 0.1 every 10 epochs. Finally, we fix the (now ternary) $\mathbf{W}_b$ and $\mathbf{W}_c$ and continue training for another 10 epochs. The resulting testing accuracy is shown in Table 5.1 for different r and p, along with the corresponding reduction in the number of multiplications compared to the uncompressed model in Table 5.2 (for the 32 × 32 CIFAR-10 images; see Table 5.D.1 in Appendix 5.D for the reduction in the number of additions). Additional results for a similar experiment based on the VGG-inspired 7-layer architecture considered in (Courbariaux et al., 2015; Li et al., 2016) can be found in Appendix 5.A.

Table 5.1: Testing accuracy (in %) of ST-ResNet-20 on CIFAR-10.

testing accuracy				
	r			
p	c_{out}	$\frac{3}{4}c_{\text{out}}$	$\frac{1}{2}c_{\text{out}}$	$\frac{1}{4}c_{\text{out}}$
1	91.24	90.62	88.63	85.46
2	89.87	89.47	87.31	84.01
4	86.13	84.67	82.67	75.01

Table 5.2: Reduction in the number of multiplications (in %) obtained by our method for ResNet-20 on CIFAR-10.

	red. in multiplications			
	r			
p	c_{out}	$\frac{3}{4}c_{\text{out}}$	$\frac{1}{2}c_{\text{out}}$	$\frac{1}{4}c_{\text{out}}$
1	98.96	99.08	99.21	99.33
2	99.33	99.36	99.39	99.42
4	99.42	99.43	99.44	99.44

Discussion: The model obtained for the base configuration with $r = c_{\text{out}}$ and $p = 1$ incurs a negligible accuracy loss compared to the uncompressed ResNet-20 with an accuracy of 91.25% (He et al., 2016a) while reducing the number of multiplications by 98.96% (the evaluation of the uncompressed ResNet-20 requires 41.038M multiply-adds). This model also matches the accuracy of TTQ (Zhu et al., 2016) for ResNet-20 while requiring fewer multiplications (TTQ does not quantize the first convolution layer). As r decreases and/or p increases, the number of multiplications decreases at the cost of further accuracy reduction.

ImageNet

We consider ST-ResNet-18 and, unlike for the experiment on CIFAR-10, we also compress the last (fully connected) layer of ST-ResNet-18 for models with $r \leq c_{\text{out}}$ in convolution layers, setting $r = 1000$ for that layer throughout (we observed that compressing the last layer when $r > c_{\text{out}}$ in convolution layers leads to a considerable reduction in validation accuracy). Following (Rastegari et al., 2016; Li et al., 2016; Zhu et al., 2016), the training images are resized such that the shorter side has length 256 and are then randomly cropped to 224×224 pixels. The validation accuracy is computed from center crops. We use an initial learning rate of 0.05 and mini-batch size 256, with two different learning rate schedules depending on the value of r

in the convolution layers: We train for 40 epochs without quantization, multiplying the learning rate by 0.1 after 30 epochs, if $r \leq c_{\text{out}}$, and for 70 epochs, multiplying the learning rate by 0.1 after 40 and 60 epochs, otherwise. Thereafter, we activate quantization and continue training for 10 epochs. Finally, we fix $\mathbf{W}_b$ and $\mathbf{W}_c$ and train $\tilde{\mathbf{a}}$ for another 5 epochs.

In Table 5.3 we report the validation accuracy of ST-ResNet-18 for different r, p, and g, and the validation accuracy obtained with KD. Table 5.4 shows the reduction in the number of multiplications compared to the original ResNet-18 model, for different r, p, and g (see Table 5.D.3 in Appendix 5.D for reductions in the number of additions and model size). In Figure 5.1, we plot the accuracy of ST-ResNet-18 for different r, p, and g, as a function of the number of operations and model size. In addition, we report the validation accuracy for related works (Rastegari et al., 2016; Li et al., 2016; Zhu et al., 2016; Lin et al., 2017) (see also Table 5.D.4 in Appendix 5.D). We do not consider $p > 2$ as this leads to (ternary) convolution with impractically large kernels for 224×224 images.

Finally, to demonstrate amenability of our method to larger models, we trained ST-ResNet-34 with $r = 2c_{\text{out}}$, $p = 2$, $g = 1$ (without tuning any hyper parameters) and obtained 69.2%/88.5% top-1/top-5 validation accuracy without KD and 71.9%/90.5% with KD (the full-precision model obtains 73.3%/91.3%; we report the accuracies of the Torch pretrained models for all full-precision ResNets).

Discussion: All ST-ResNet-18 models that require the same number of multiplications as TWN (those with $r = c_{\text{out}}$, $p = 1$, $g = 4$; $r = c_{\text{out}}$, $p = 1$, $g = 1$; $r = 2c_{\text{out}}$, $p = 2$, $g = 1$) obtain a considerably higher top-1 and top-5 accuracy than TWN. In particular, ST-ResNet-18 with $r = 2c_{\text{out}}$, $p = 2$, and $g = 1$ leads to a 7.0% improvement in top-1 accuracy. Furthermore, ST-ResNet-18 with $r = 2c_{\text{out}}$, $p = 1$, and $g = 1$ outperforms TTQ while using 98.3% fewer multiplications. ST-ResNet-18 with $r = 6c_{\text{out}}$, $p = 2$, and $g = 1$ incurs a 2.0% reduction in top-1 accuracy compared to the full-precision model while reducing the number of multiplications by 99.63%. Our ST-ResNets require fewer

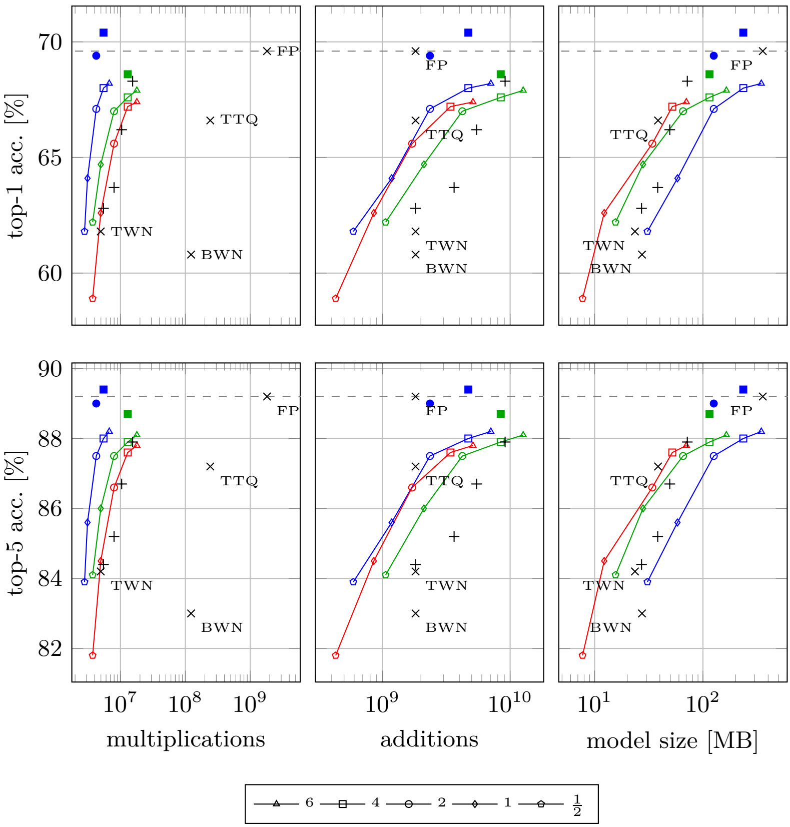


Fig. 5.1: Top-1 and top-5 validation accuracy of ST-ResNet-18 on ImageNet as a function of the number of multiplications, the number of additions, and model size, along with the values obtained in related works BWN (Rastegari et al., 2016), TWN (Li et al., 2016), TTQ (Zhu et al., 2016), ABC-Net-1/2/3/5 (Lin et al., 2017) ("+" signs, the suffix reflects the ranking according to accuracy), and the full-precision model (FP). The numbers associated with the marker types correspond to the ratio of the number of hidden SP units and output channels, r/c_{out}. Different colors indicate different combinations of output patch size p and number of convolution groups g: Blue: $p = 2$, $g = 1$; green: $p = 1$, $g = 1$; red: $p = 1$, $g = 4$. Selected models trained with KD are shown with filled markers.

Table 5.3: Top-1 and top-5 validation accuracy (in %) of ST-ResNet-18 on ImageNet, for different choices of r, p, g, and with KD.

top-1 accuracy

	r					r (KD)		
(p,g)	$6c_{\text{out}}$	$4c_{\text{out}}$	$2c_{\text{out}}$	c_{out}	$\frac{1}{2}c_{\text{out}}$	$4c_{\text{out}}$	$2c_{\text{out}}$	c_{out}
$(1,1)$	67.9	67.6	67.0	64.7	62.2	68.6	67.9	66.0
$(2,1)$	68.2	68.0	67.1	64.1	61.8	70.4	69.4	66.4
$(1,4)$	67.4	67.2	65.6	62.6	58.9	68.0	66.6	63.9

top-5 accuracy

	r					r (KD)		
(p,g)	$6c_{\text{out}}$	$4c_{\text{out}}$	$2c_{\text{out}}$	c_{out}	$\frac{1}{2}c_{\text{out}}$	$4c_{\text{out}}$	$2c_{\text{out}}$	c_{out}
$(1,1)$	88.1	87.9	87.5	86.0	84.1	88.7	88.3	87.1
$(2,1)$	88.2	88.0	87.5	85.6	83.9	89.4	89.0	87.3
$(1,4)$	87.8	87.6	86.6	84.5	81.8	88.3	87.5	85.5

Table 5.4: Reduction in the number of multiplications (in %) of ST-ResNet-18 compared to the full-precision model, for 224×224 images.

red. in multiplications

	r				
(p,g)	$6c_{\text{out}}$	$4c_{\text{out}}$	$2c_{\text{out}}$	c_{out}	$\frac{1}{2}c_{\text{out}}$
$(1,1)$	99.01	99.29	99.56	99.73	99.79
$(2,1)$	99.63	99.70	99.77	99.83	99.85
$(1,4)$	99.01	99.29	99.56	99.73	99.79

multiplications and additions than ABC-Net-1/2/3 while yielding the same accuracy. For $p = 2$, our models lead to a reduction in multiplications of at least 50% compared to the ABC-Net with the same accuracy. Note that TTQ and BWN use considerably more multiplications than ST-ResNet-18, TWN, and ABC-Net as they do not quantize the first convolution layer.

In contrast to the experiments on CIFAR-10, increasing p from 1 to 2 increases the accuracy for fixed $r \geq 2c_{\text{out}}$. A possible explanation for this behavior is that the benefits of the increase in the number of ternary parameters obtained by increasing p outweighs the loss in precision due to the reduction in spatial resolution. This is in accordance with the fact that the images in ImageNet are much larger than those in CIFAR-10, resulting in larger feature maps for most layers.

KD leads to improvements in top-1 accuracy of 1.3–3.5%, see Table 5.3. In particular, ST-ResNet-18 with $r = 2c_{\text{out}}$, $p = 2$, and $g = 1$ trained using KD essentially matches the accuracy of the full-precision model. Increasing r to $4c_{\text{out}}$ yields a model that even outperforms the full-precision model. To the best of our knowledge, these models are the first to realize massive reductions in the number of multiplications (over 99.5%) while maintaining the accuracy of the full-precision model. Note that student models outperforming their teachers were observed before in different contexts (Mishra and Marr, 2018; Zhuang et al., 2018; Furlanello et al., 2018).

For some of the configurations, the reduction in multiplications comes at the cost of a small to moderate increase in the number of additions. We emphasize that this is also the case for Strassen's algorithm (see (5.2)) and the Winograd filter-based convolution (see (Lavin and Gray, 2016, Sec. 4.1)). The specific application and target platform will determine what increase in the number of additions is acceptable.

Finally, in all our image classification experiments the ratio r/c_{out} is the same for all layers. Since one would expect improvements from allocating more multiplications to layers that require more accurate operations, we also tested a simple way to learn the ratio r/c_{out} for each layer from data. Specifically, we chose a large r/c_{out} and applied L1 regularization to the vectors $\tilde{\mathbf{a}}$. However, for a given total multiplication budget this strategy led to lower accuracy in our experiments than just fixing r/c_{out} for all layers.

5.3.3. Language modeling

We apply our method to the character-level language model described in (Kim et al., 2016a) and evaluate it on the English Penn Treebank (PTB with word vocabulary size 10k, character vocabulary size 51, 1M training tokens, standard train-validation-test split, see (Kim et al., 2016a)) (Marcus et al., 1993). We use the large version of the model from (Kim et al., 2016a) which is composed of a convolution layer with 1100 filters (applied to a character-level representation of the words, without aggregation across channels), followed by a 2-layer highway network with 1100 hidden units, feeding into a 2-layer LSTM network with 650 hidden units (see Table 2 in (Kim et al., 2016a) for more details). We obtain Strassen language models (ST-LMs) by replacing the convolution layer and all fully connected layers (both within the LSTM and the output/decode layer) with SPNs. r is set to the number of filters for the convolution layer and is parametrized as $r(\kappa) = \kappa \cdot n_{\mathrm{out}}$ for the fully connected layers, where n_{out} is the number of hidden units. For the output/decode layer we use $r(\kappa) = \kappa \cdot 2000$.

All models are trained for 40 epochs using SGD with mini-batch size 20 and initial learning rate 2, multiplying the learning rate by 0.5 when the validation perplexity per word (PPL; c.f. (Kim et al., 2016a, Eq. (9))) decreases by less than 0.5 per epoch (a similar schedule was used in (Kim et al., 2016a)). Although the ST-LMs train stably with quantization from scratch, we train them for 20 epochs with full-precision weights before activating quantization for $\mathbf{W}_b$ and $\mathbf{W}_c$, which leads to slightly lower validation PPLs. As a baseline, we consider the TWN quantization scheme (5.4) and apply it to all layers of the language model. As we observed a somewhat higher variability in the validation performance than for the image classification experiments, we train each quantized model 5 times and report the average testing PPL.

In Figure 5.2, we plot the average testing PPL of our ST-LMs for different r as a function of the number of operations and model size, with and without KD. Table 5.D.2 in Appendix 5.D shows the reduction in the number of operations and model size compared to the full-precision model.

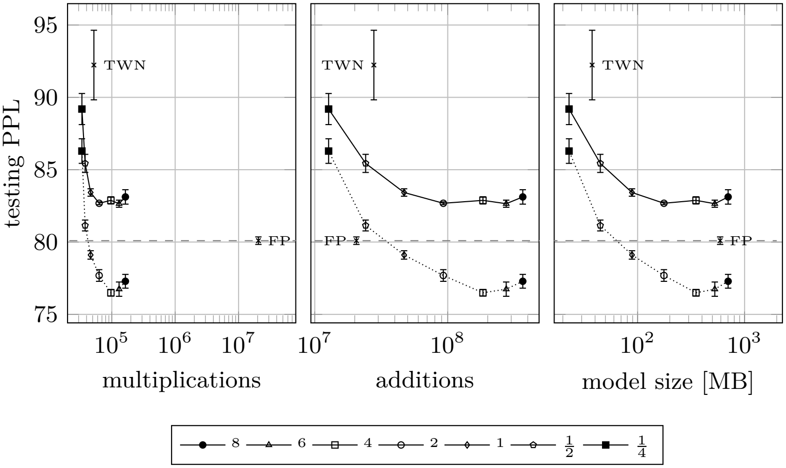


Fig. 5.2: Testing PPL (averaged over 5 runs) for ST-LM as a function of the number of operations and model size, along with the values obtained for TWN quantization (5.4), and the full-precision model (FP). Solid line: Without KD; dotted line: With KD. The numbers associated with the marker types correspond to the ratio of the number of hidden SP units and hidden units, r/n_{out}.

Discussion: Our ST-LM models reduce the number of multiplications by over 99% compared to the full-precision model, while incurring an increase in testing PPL of only 3–4%. The PPL obtained via TWN quantization clearly exceeds that of all considered ST-LMs. The ST-LM model with $r = n_{\text{out}}$ requires roughly the same number of multiplications as the TWN model but has a 7.4% lower testing PPL. KD leads to a significant reduction in testing PPL. The distilled ST-LMs outperform the teacher model for $r \geq n_{\text{out}}$. To our knowledge, our models are the first to obtain such massive reductions (over 99.5% for $r \leq 4n_{\text{out}}$) in the number of multiplications while maintaining the PPL of the full-precision model. We observed that KD applied to the teacher model also reduces its testing PPL to values comparable to that of the compressed models with KD for $r \geq n_{\text{out}}$ (see (Furlanello et al., 2018) for more exploration of this phenomenon). On the other hand, KD considerably increases the testing PPL for TWN.

There are only few prior works on compressing sequence models in a *multiplication-reducing* fashion (He et al., 2016b; Hubara et al., 2016). For single-layer LSTM and GRU language models with binary weights He et al. (2016b) report an increase in PPL of 70% and more compared to the full-precision model, whereas Hubara et al. (2016) observed divergence for a single-layer LSTM model with binary weights, but report small degradations for 4-bit weight and activation quantization.

5.4. CONCLUSION AND FUTURE WORK

We proposed and evaluated a versatile framework to learn fast approximate matrix multiplications for DNNs end-to-end. We found that our method leads to the same or higher accuracy compared to existing methods while using significantly fewer multiplications. By leveraging KD we were able to train models that incur no loss in predictive performance despite reducing the number of multiplications by over 99.5%. A natural next step is to incorporate activation quantization into the proposed method. In addition, it will be interesting to see how the theoretical gains reported here translate into actual energy savings and runtime speedups on specialized hardware such as FPGAs and ASICs.

APPENDICES

5.A. ADDITIONAL RESULTS ON CIFAR-10

We apply our method to the 7-layer VGG-inspired architecture previously considered in (Courbariaux et al., 2015; Li et al., 2016) (see (Li et al., 2016, Sec. 3)) for a detailed description of the architecture) and evaluate it on CIFAR-10. We vary r and p for convolution layers and fix $r = 1024$ for the fully connected layer with 1024 units feeding into the softmax layer. The same hyper parameters and schedules as for ST-ResNet-20 are used for training, see Sec. 5.3.2. Table 5.A.1 shows the testing accuracy for different r and p. Our method achieves the same or higher accuracy than TWN (Li et al., 2016) for $p = 1$. The impact of increasing p on testing accuracy is analogous to that observed for ST-ResNet-20. Reducing r seems to reduce testing accuracy to a smaller extent than for ST-ResNet-20. A possible reason for this could be that the considered VGG architecture has considerably wider layers (128 to 512 channels) than ResNet-20 (16 to 64 channels).

Table 5.A.1: Testing accuracy (in %) of the 7-layer VGG model from (Courbariaux et al., 2015; Li et al., 2016) compressed by our method, on CIFAR-10.

testing accuracy

	r			
p	c_{out}	$\frac{3}{4}c_{\text{out}}$	$\frac{1}{2}c_{\text{out}}$	$\frac{1}{4}c_{\text{out}}$
1	93.17	93.19	92.39	92.50
2	91.87	91.44	91.39	89.66
4	88.08	88.40	87.65	87.05

5.B. ADDITIONAL RESULTS FOR LANGUAGE MODELING

To assess the generalization of the ST-LM models described in Section 5.3.3, we apply the FP and ST-LM (with $r = 2n_{\text{out}}$) models to Wikitext-2[3] (word vocabulary size 33k and 2M training tokens) without changing hyper parameters and obtain a testing PPL of 90.07, 97.40, and 87.72 for the FP model, the ST-LM model, and the ST-LM model with KD, respectively. The PPL of the FP model (19M parameters) is comparable to that of the variational dropout LSTM (VD-LSTM-RE, 22M parameters) from (Inan et al., 2017). While the gap between the FP and ST-LM model is larger than for PTB, the ST-LM model with distillation outperforms the FP model similarly as for PTB.

5.C. APPLICATION TO 2D CONVOLUTION: PSEUDOCODE

In this section, we provide provide pseudocode to facilitate the implementation of the proposed framework for 2D convolutions with $k \times k$ kernels (see Section 5.2.4) in popular deep learning software packages. Let `W_B`, `a_tilde`, and `W_C` be variables associated with tensors of dimensions $r \times c_{\text{in}} \times (p-1+k) \times (p-1+k)$, $1 \times r \times 1 \times 1$, and $r \times c_{\text{out}} \times p \times p$, respectively. Denote the standard 2D convolution and transposed 2D convolution operations with input `data`, filter tensor `weights`, `in_channels` input channels, `out_channels` output channels, kernel size `kernel_size`, and stride `stride` by `Conv2d` and `ConvTranspose2d`. Let `Multiply` be the broadcasted element-wise multiplication of `weights` with `data` and designate the function implementing the quantization scheme described in (5.4) by `Quantize`. Then, the forward pass (during training) through a compressed 2D

[3] Available at `https://www.salesforce.com/products/einstein/ai-research/the-wikitext-dependency-language-modeling-dataset/`.

convolution for an input tensor `in_data` of dimensions $b \times c_{\text{in}} \times W \times H$ is given by

```
W_B = Quantize(W_B)
W_C = Quantize(W_C)
conv_out = Conv2d(
               data=in_data,
               weights=W_B,
               in_channels=$c_{\text{in}}$,
               out_channels=$r$,
               kernel_size=$p - 1 + k$,
               stride=$p$,
               groups=$g$)
mul_out = Multiply(
               data=conv_out,
               weights=a_tilde)
out_data = ConvTranspose2d(
               data=mul_out,
               weights=W_C,
               in_channels=$r$,
               out_channels=$c_{\text{out}}$,
               kernel_size=$p$,
               stride=$p$)
```

At inference time, `Conv2d` and `ConvTranspose2d` can be replaced with specialized convolution operations exploiting the fact that `W_B` and `W_C` are ternary. To compute the backward pass, the backpropagation algorithm (Rumelhart et al., 1986) is applied to the sequence of operations in the forward pass, ignoring the `Quantize` operations. We found it beneficial to perform batch normalization (Ioffe and Szegedy, 2015) after the `Conv2d` operation.

5.D. ADDITIONAL TABLES

Table 5.D.1: Reduction (in %) in the number of additions obtained by our method for ResNet-20 on CIFAR-10.

red. in additions

	r			
p	c_{out}	$\frac{3}{4}c_{\text{out}}$	$\frac{1}{2}c_{\text{out}}$	$\frac{1}{4}c_{\text{out}}$
1	-13.904	14.435	42.774	71.112
2	39.123	54.205	69.287	84.369
4	57.550	68.025	78.501	88.976

Table 5.D.2: Testing PPL and reduction (in %) in the number of operations as well as model size for ST-LM compared to the full-precision model. We also report the reductions for the model compressed with TWN quantization (5.4).

	ST-LM, r							
	$8n_{\text{out}}$	$6n_{\text{out}}$	$4n_{\text{out}}$	$2n_{\text{out}}$	n_{out}	$\frac{1}{2}n_{\text{out}}$	$\frac{1}{4}n_{\text{out}}$	TWN
testing PPL	83.118	82.65	82.88	82.68	83.42	85.44	89.19	92.23
testing PPL (dist.)	77.29	76.74	76.49	77.69	79.11	81.14	86.29	-
multiplications	99.20	99.37	99.53	99.69	99.77	99.82	99.84	99.75
additions	-1682.23	-1238.24	-794.28	-350.31	-128.33	-17.34	38.21	-35.27
model size	-18.76	10.89	40.54	70.19	85.01	92.42	96.13	93.65

Table 5.D.3: Reduction in the number of additions and model size (in %) of ST-ResNet-18 compared to the full-precision model, for 224×224 images.

red. in additions

	r				
(p, g)	$6c_{\text{out}}$	$4c_{\text{out}}$	$2c_{\text{out}}$	c_{out}	$\frac{1}{2}c_{\text{out}}$
$(1, 1)$	-596.76	-364.56	-132.36	-16.32	41.73
$(2, 1)$	-288.57	-159.10	-29.63	35.05	67.41
$(1, 4)$	-181.51	-87.73	6.05	52.89	76.33

red. in model size

	r				
(p, g)	$6c_{\text{out}}$	$4c_{\text{out}}$	$2c_{\text{out}}$	c_{out}	$\frac{1}{2}c_{\text{out}}$
$(1, 1)$	53.87	67.76	81.64	92.16	95.63
$(2, 1)$	3.07	33.89	64.71	83.69	91.40
$(1, 4)$	80.31	85.38	90.45	96.56	97.83

Table 5.D.4: Top-1 and top-5 validation accuracy (in %) along with the reduction (in %) in the number of multiplications and model size for BWN (Rastegari et al., 2016), TWN (Li et al., 2016), TTQ (Zhu et al., 2016), ABC-Net-3 (Lin et al., 2017), and ST-ResNet-18-2-2-1 ($r = 2c_{\text{out}}$, $p = 2$, $g = 1$), compared to the full-precision model (for the full-precision model, the absolute quantities are given in parentheses).

	top-1	top-5	mul.	model size
BWN	60.8	83.0	93.25	92.33
TWN	61.8	84.2	99.73	93.39
TTQ	66.6	87.2	86.66	89.20
ABC-Net-3	66.2	86.7	99.42	49.39
ST-ResNet-18-2-2-1	67.1	87.5	99.77	125.89
full-precision	69.6	89.2	$(1.82 \cdot 10^9)$	(356.74 MB)

References

Adamczak, R. (2015), “A note on the Hanson-Wright inequality for random vectors with dependencies,” *Electronic Communications in Probability*, vol. 20, no. 72, pp. 1–13.

Agustsson, E., Mentzer, F., Tschannen, M., Cavigelli, L., Timofte, R., Benini, L., and Gool, L. V. (2017), “Soft-to-hard vector quantization for end-to-end learning compressible representations,” in *Advances in Neural Information Processing Systems (NIPS)*, pp. 1141–1151.

Agustsson, E., Tschannen, M., Mentzer, F., Timofte, R., and Van Gool, L. (2018), “Generative adversarial networks for extreme learned image compression,” *arXiv:1804.02958*.

Andri, R., Cavigelli, L., Rossi, D., and Benini, L. (2018), “YodaNN: An architecture for ultralow power binary-weight CNN acceleration,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 1, pp. 48–60.

Andrzejak, R. G., Lehnertz, K., Mormann, F., Rieke, C., David, P., and Elger, C. E. (2001), “Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state,” *Physical Review E*, vol. 64, no. 6.

Arjovsky, M., Chintala, S., and Bottou, L. (2017), “Wasserstein generative adversarial networks,” in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 214–223.

Balcan, M.-F., Blum, A., and Vempala, S. (2008), “A discriminative framework for clustering via similarity functions,” in *Proceedings of the Annual ACM Symposium on Theory of Computing*, pp. 671–680.

Ballé, J., Laparra, V., and Simoncelli, E. P. (2017), “End-to-end optimized image compression,” in *International Conference on Learning Representations (ICLR)*.

Ballé, J., Minnen, D., Singh, S., Hwang, S. J., and Johnston, N. (2018), "Variational image compression with a scale hyperprior," in *International Conference on Learning Representations (ICLR)*.

Basri, R. and Jacobs, D. (2003), "Lambertian reflectance and linear subspaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 218–233.

Bellard, F. (2018), "BPG Image format," `https://bellard.org/bpg/`. Accessed 26 June 2018.

Blumensath, T., Davies, M. E., and Rilling, G. (2012), "Greedy algorithms for compressed sensing," in Y. C. Eldar and G. Kutyniok (Eds.), *Compressed Sensing: Theory and Applications*, pp. 348–393, Cambridge University Press.

Boets, J., De Cock, K., De Moor, B., and Espinoza, M. (2005), "Clustering time series, subspace identification and cepstral distances," *Communications in Information & Systems*, vol. 5, no. 1, pp. 69–96.

Borysov, P., Hannig, J., and Marron, J. (2014), "Asymptotics of hierarchical clustering for growing dimension," *Journal of Multivariate Analysis*, vol. 124, pp. 465–479.

Boucheron, S., Lugosi, G., and Massart, P. (2013), *Concentration Inequalities: A Nonasymptotic Theory of Independence*, Oxford University Press.

Bucilua, C., Caruana, R., and Niculescu-Mizil, A. (2006), "Model compression," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 535–541.

Buldygin, V. and Moskvichova, K. (2013), "The sub-Gaussian norm of a binary random variable," *Theory of Probability and Mathematical Statistics*, vol. 86, pp. 33–49.

Cai, T. T. and Wang, L. (2011), "Orthogonal matching pursuit for sparse signal recovery with noise," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4680–4688.

Caiado, J., Crato, N., and Peña, D. (2006), "A periodogram-based metric for time series classification," *Computational Statistics & Data Analysis*, vol. 50, no. 10, pp. 2668–2684.

Chen, S., Billings, S. A., and Luo, W. (1989), "Orthogonal least squares methods and their application to non-linear system identification," *International Journal of Control*, vol. 50, no. 5, pp. 1873–1896.

Chen, W.-Y., Song, Y., Bai, H., Lin, C.-J., and Chang, E. Y. (2011), "Parallel spectral clustering in distributed systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 568–586.

Chikuse, Y. (2003), *Statistics on special manifolds*, *Lecture Notes in Statistics*, vol. 174, Springer Science & Business Media.

Cohen, T. and Welling, M. (2016), “Group equivariant convolutional networks,” in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 2990–2999.

Corduas, M. and Piccolo, D. (2008), “Time series clustering and classification by the autoregressive metric,” *Computational Statistics & Data Analysis*, vol. 52, no. 4, pp. 1860–1872.

Costeira, J. P. and Kanade, T. (1998), “A multibody factorization method for independently moving objects,” *International Journal of Computer Vision*, vol. 29, no. 3, pp. 159–179.

Courbariaux, M., Bengio, Y., and David, J.-P. (2015), “Binaryconnect: Training deep neural networks with binary weights during propagations,” in *Advances in Neural Information Processing Systems (NIPS)*, pp. 3123–3131.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., and Wei, Y. (2017), “Deformable convolutional networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 764–773.

Delp, E. J. and Mitchell, O. R. (1991), “Moment preserving quantization (signal processing),” *IEEE Transactions on Communications*, vol. 39, no. 11, pp. 1549–1558.

Demanet, L., Létourneau, P.-D., Boumal, N., Calandra, H., Chiu, J., and Snelson, S. (2012), “Matrix probing: A randomized preconditioner for the wave-equation Hessian,” *Applied and Computational Harmonic Analysis*, vol. 32, no. 2, pp. 155–168.

Denton, E. L., Zaremba, W., Bruna, J., LeCun, Y., and Fergus, R. (2014), “Exploiting linear structure within convolutional networks for efficient evaluation,” in *Advances in Neural Information Processing Systems (NIPS)*, pp. 1269–1277.

Donahue, J., Krähenbühl, P., and Darrell, T. (2017), “Adversarial feature learning,” in *International Conference on Learning Representations (ICLR)*.

Dosovitskiy, A. and Brox, T. (2016), “Generating images with perceptual similarity metrics based on deep networks,” in *Advances in Neural Information Processing Systems (NIPS)*, pp. 658–666.

Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., and Courville, A. (2017), “Adversarially learned inference,” in *International Conference on Learning Representations (ICLR)*.

Dyer, E. L., Sankaranarayanan, A. C., and Baraniuk, R. G. (2013), “Greedy feature selection for subspace clustering,” *Journal of Machine Learning Research*, vol. 14, pp. 2487–2517.

Elhamifar, E. and Vidal, R. (2013), “Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781.

Elser, V. (2016), "A network that learns Strassen multiplication," *Journal of Machine Learning Research*, vol. 17, no. 116, pp. 1–13.

Esling, P. and Agon, C. (2012), "Time-series data mining," *ACM Computing Surveys*, vol. 45, no. 1, p. 12.

Ferreira, L. N. and Zhao, L. (2016), "Time series clustering via community detection in networks," *Information Sciences*, vol. 326, pp. 227–242.

Foucart, S. and Rauhut, H. (2013), *A Mathematical Introduction to Compressive Sensing*, Springer, Berlin, Heidelberg.

Friedman, J., Hastie, T., and Tibshirani, R. (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer New York.

Friedman, J. H. and Stuetzle, W. (1981), "Projection pursuit regression," *Journal of the American Statistical Association*, vol. 76, no. 376, pp. 817–823.

Furlanello, T., Lipton, Z. C., Tschannen, M., Itti, L., and Anandkumar, A. (2018), "Born-again neural networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1602–1611.

Galteri, L., Seidenari, L., Bertini, M., and Del Bimbo, A. (2017), "Deep generative adversarial compression artifact removal," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 4826–4835.

Georghiades, A. S., Belhumeur, P. N., and Kriegman, D. J. (2001), "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660.

Golub, G. H. and Van Loan, C. F. (1996), *Matrix Computations*, JHU Press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014), "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 2672–2680.

Graf, S. and Luschgy, H. (2007), *Foundations of quantization for probability distributions*, Springer.

Gray, R. M. (2006), "Toeplitz and circulant matrices: A review," *Foundations & Trends in Communications and Information Theory*, vol. 2, no. 3, pp. 155–239.

——— (2009), *Probability, Random Processes, and Ergodic Properties*, Springer Science & Business Media.

Gregor, K., Besse, F., Rezende, D. J., Danihelka, I., and Wierstra, D. (2016), "Towards conceptual compression," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 3549–3557.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012), "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, pp. 723–773.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017), "Improved training of Wasserstein GANs," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 5769–5779.

He, K., Zhang, X., Ren, S., and Sun, J. (2016a), "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

He, Q., Wen, H., Zhou, S., Wu, Y., Yao, C., Zhou, X., and Zou, Y. (2016b), "Effective quantization methods for recurrent neural networks," *arXiv:1611.10176*.

Heckel, R. and Bölcskei, H. (2015), "Robust subspace clustering via thresholding," *IEEE Transactions on Information Theory*, vol. 61, no. 11, pp. 6320–6342.

Heckel, R., Tschannen, M., and Bölcskei, H. (2017), "Dimensionality-reduced subspace clustering," *Information and Inference: A Journal of the IMA*, vol. 6, no. 3, pp. 246–283.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017), "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 6629–6640.

Hinton, G., Vinyals, O., and Dean, J. (2014), "Distilling the knowledge in a neural network," *NIPS Deep Learning Workshop*.

Ho, J., Yang, M., Lim, J., Lee, K., and Kriegman, D. (2003), "Clustering appearances of objects under varying illumination conditions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 11–18.

Hong, W., Wright, J., Huang, K., and Ma, Y. (2006), "Multiscale hybrid linear models for lossy image representation," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3655–3671.

Horn, R. A. and Johnson, C. R. (1991), *Topics in Matrix Analysis*, Cambridge University Press.

Horowitz, M. (2014), "Computing's energy problem (and what we can do about it)," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pp. 10–14.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017), "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv:1704.04861*.

Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., and Bengio, Y. (2016), "Quantized neural networks: Training neural networks with low precision weights and activations," *arXiv:1609.07061*.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., and Keutzer, K. (2016), "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and $<$ 0.5 MB model size," *arXiv:1602.07360*.

Inan, H., Khosravi, K., and Socher, R. (2017), "Tying word vectors and word classifiers: A loss framework for language modeling," in *International Conference on Learning Representations (ICLR)*.

Ioffe, S. and Szegedy, C. (2015), "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 448–456.

Jain, P., Tewari, A., and Dhillon, I. S. (2011), "Orthogonal matching pursuit with replacement," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 1215–1223.

Jiang, D., Tang, C., and Zhang, A. (2004), "Cluster analysis for gene expression data: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 11, pp. 1370–1386.

Johnston, N., Vincent, D., Minnen, D., Covell, M., Singh, S., Chinen, T., Jin Hwang, S., Shor, J., and Toderici, G. (2017), "Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks," *arXiv:1703.10114*.

Kakizawa, Y., Shumway, R. H., and Taniguchi, M. (1998), "Discrimination and clustering for multivariate time series," *Journal of the American Statistical Association*, vol. 93, no. 441, pp. 328–340.

Kalpakis, K., Gada, D., and Puttagunta, V. (2001), "Distance measures for effective clustering of ARIMA time-series," in *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pp. 273–280.

Kankanahalli, S. (2018), "End-to-end optimized speech coding with deep neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2521–2525.

Katsavounidis, I., Jay Kuo, C.-C., and Zhang, Z. (1994), "A new initialization technique for generalized Lloyd iteration," *IEEE Signal Processing Letters*, vol. 1, no. 10, pp. 144–146.

Kay, S. M. (1988), *Modern Spectral Estimation*, Prentice Hall.

Khaleghi, A., Ryabko, D., Mary, J., and Preux, P. (2012), "Online clustering of processes," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 601–609.

——— (2016), “Consistent algorithms for clustering time series,” *Journal of Machine Learning Research*, vol. 17, no. 3, pp. 1–32.

Kim, Y., Jernite, Y., Sontag, D., and Rush, A. M. (2016a), “Character-aware neural language models,” in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 2741–2749.

Kim, Y. and Rush, A. M. (2016), “Sequence-level knowledge distillation,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1317–1327.

Kim, Y.-D., Park, E., Yoo, S., Choi, T., Yang, L., and Shin, D. (2016b), “Compression of deep convolutional neural networks for fast and low power mobile applications,” in *International Conference on Learning Representations (ICLR)*.

Kingma, D. P. and Ba, J. (2015), “Adam: A method for stochastic optimization,” in *International Conference on Learning Representations (ICLR)*.

Kingma, D. P. and Welling, M. (2014), “Auto-encoding variational Bayes,” in *International Conference on Learning Representations (ICLR)*.

Krizhevsky, A. and Hinton, G. (2009), “Learning multiple layers of features from tiny images,” .

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012), “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems (NIPS)*, pp. 1097–1105.

Larsen, A. B. L., Sønderby, S. K., Larochelle, H., and Winther, O. (2015), “Autoencoding beyond pixels using a learned similarity metric,” *arXiv:1512.09300*.

Laurent, B. and Massart, P. (2000), “Adaptive estimation of a quadratic functional by model selection,” *The Annals of Statistics*, vol. 28, no. 5, pp. 1302–1338.

Lavin, A. and Gray, S. (2016), “Fast algorithms for convolutional neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4013–4021.

Lebedev, V. and Lempitsky, V. (2016), “Fast convnets using group-wise brain damage,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2554–2564.

Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W. (2017), “Photo-realistic single image super-resolution using a generative adversarial network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4681–4690.

Ledoux, M. (2005), *The concentration of measure phenomenon*, no. 89 in Mathematical Surveys and Monographs, American Mathematical Soc.

Lee, K. C., Ho, J., and Kriegman, D. J. (2005), "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 684–698.

Li, F., Zhang, B., and Liu, B. (2016), "Ternary weight networks," *NIPS Workshop on Efficient Methods for Deep Neural Networks (EMDNN).*

Li, L. and Prakash, B. A. (2011), "Time series clustering: Complex is simpler!" in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 185–192.

Li, M., Klejsa, J., and Kleijn, W. B. (2010), "Distribution preserving quantization with dithering and transformation," *IEEE Signal Processing Letters*, vol. 17, no. 12, pp. 1014–1017.

Li, M., Lian, X.-C., Kwok, J. T., and Lu, B.-L. (2011), "Time and space efficient spectral clustering via column sampling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2297–2304.

Li, M., Zuo, W., Gu, S., Zhao, D., and Zhang, D. (2018), "Learning convolutional networks for content-weighted image compression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3214–3223.

Liese, F. and Miescke, K.-J. (2007), "Statistical decision theory," in *Statistical Decision Theory*, pp. 1–52, Springer.

Lin, X., Zhao, C., and Pan, W. (2017), "Towards accurate binary convolutional neural network," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 344–352.

Liu, B., Wang, M., Foroosh, H., Tappen, M., and Pensky, M. (2015a), "Sparse convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 806–814.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015b), "Deep learning face attributes in the wild," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3730–3738.

Locatello, F., Khanna, R., Tschannen, M., and Jaggi, M. (2017a), "A unified optimization view on generalized matching pursuit and frank-wolfe," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 860–868.

Locatello, F., Tschannen, M., Rätsch, G., and Jaggi, M. (2017b), "Greedy algorithms for cone constrained optimization with convergence guarantees," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 773–784.

Lucic, M., Kurach, K., Michalski, M., Gelly, S., and Bousquet, O. (2018), "Are GANs Created Equal? A Large-Scale Study," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 698–707.

Luschgy, H. and Pagès, G. (2002), "Functional quantization of Gaussian processes," *Journal of Functional Analysis*, vol. 196, no. 2, pp. 486–531.

von Luxburg, U. (2007), "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416.

Maharaj, E. A. and D'Urso, P. (2011), "Fuzzy clustering of time series in the frequency domain," *Information Sciences*, vol. 181, no. 7, pp. 1187–1211.

Mallat, S. G. and Zhang, Z. (1993), "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993), "Building a large annotated corpus of English: The Penn Treebank," *Computational Linguistics*, vol. 19, no. 2, pp. 313–330.

Marti, G., Andler, S., Nielsen, F., and Donnat, P. (2016a), "Clustering financial time series: How long is enough?" in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2583–2589.

Marti, G., Nielsen, F., and Donnat, P. (2016b), "Optimal copula transport for clustering multivariate time series," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2379–2383.

Marti, G., Nielsen, F., Donnat, P., and Andler, S. (2017), "On clustering financial time series: A need for distances between dependent random variables," in *Computational Information Geometry*, pp. 149–174, Springer.

Maruyama, G. (1949), "The harmonic analysis of stationary stochastic processes," *Memoirs of the Faculty of Science, Kyushu University. Series A, Mathematics*, vol. 4, no. 1, pp. 45–106.

Mathieu, M., Couprie, C., and LeCun, Y. (2016), "Deep multi-scale video prediction beyond mean square error," in *International Conference on Learning Representations (ICLR)*.

Mentzer, F., Agustsson, E., Tschannen, M., Timofte, R., and Van Gool, L. (2018), "Conditional probability models for deep image compression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4394–4402.

Mishra, A. and Marr, D. (2018), "Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy," in *International Conference on Learning Representations (ICLR)*.

Muirhead, R. J. (2009), *Aspects of multivariate statistical theory*, *Wiley Series in Probability and Statistics*, vol. 197, John Wiley & Sons.

Nasihatkon, B. and Hartley, R. (2011), "Graph connectivity in sparse subspace clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2137–2144.

Ng, A., Jordan, I. M., and Yair, W. (2001), "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 849–856.

Ng, E. W. and Geller, M. (1969), "A table of integrals of the error functions," *Journal of Research of the National Bureau of Standards B*, vol. 73, pp. 1–20.

Novikov, A., Podoprikhin, D., Osokin, A., and Vetrov, D. P. (2015), "Tensorizing neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 442–450.

Nurvitadhi, E., Venkatesh, G., Sim, J., Marr, D., Huang, R., Ong Gee Hock, J., Liew, Y. T., Srivatsan, K., Moss, D., Subhaschandra, S., and Boudoukh, G. (2017), "Can FPGAs beat GPUs in accelerating next-generation deep neural networks?" in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 5–14.

Park, D., Caramanis, C., and Sanghavi, S. (2014), "Greedy subspace clustering," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 2753–2761.

Polino, A., Pascanu, R., and Alistarh, D. (2018), "Model compression via distillation and quantization," *International Conference on Learning Representations (ICLR)*.

Radford, A., Metz, L., and Chintala, S. (2015), "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv:1511.06434*.

Rastegari, M., Ordonez, V., Redmon, J., and Farhadi, A. (2016), "XNOR-Net: Imagenet classification using binary convolutional neural networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 525–542.

Rippel, O. and Bourdev, L. (2017), "Real-time adaptive image compression," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 2922–2930.

Rosca, M., Lakshminarayanan, B., Warde-Farley, D., and Mohamed, S. (2017), "Variational approaches for auto-encoding generative adversarial networks," *arXiv:1706.04987*.

Rudelson, M. and Vershynin, R. (2013), "Hanson-Wright inequality and sub-

Gaussian concentration," *Electronic Communications in Probability*, vol. 18, pp. no. 82, 1–9.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986), "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, p. 533.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015), "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252.

Ryabko, D. (2010), "Clustering processes," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 919–926.

Ryabko, D. and Mary, J. (2013), "A binary-classification-based metric between time-series distributions and its use in statistical and learning problems," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2837–2856.

Sanei, S. and Chambers, J. A. (2008), *EEG Signal Processing*, John Wiley & Sons.

Santurkar, S., Budden, D., and Shavit, N. (2018), "Generative compression," in *Picture Coding Symposium (PCS)*, pp. 258–262.

Schudy, W. and Sviridenko, M. (2012), "Concentration and moment inequalities for polynomials of independent random variables," in *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 437–446.

Skinner, D. (1976), "Pruning the decimation in-time FFT algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 2, pp. 193–194.

Soltanolkotabi, M. and Candès, E. J. (2012), "A geometric analysis of subspace clustering with outliers," *The Annals of Statistics*, vol. 40, no. 4, pp. 2195–2238.

Soltanolkotabi, M., Elhamifar, E., and Candès, E. J. (2014), "Robust subspace clustering," *The Annals of Statistics*, vol. 42, no. 2, pp. 669–699.

Stoica, P. and Moses, R. L. (2005), *Spectral Analysis of Signals*, Pearson/Prentice Hall, Upper Saddle River, NJ.

Strassen, V. (1969), "Gaussian elimination is not optimal," *Numerische Mathematik*, vol. 13, no. 4, pp. 354–356.

Sze, V., Chen, Y.-H., Yang, T.-J., and Emer, J. S. (2017), "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329.

Tai, C., Xiao, T., Zhang, Y., and Wang, X. (2016), "Convolutional neural networks with low-rank regularization," in *International Conference on Learning Representations (ICLR)*.

Temlyakov, V. N. (2003), "Nonlinear methods of approximation," *Foundations of Computational Mathematics*, vol. 3, no. 1, pp. 33–107.

Theis, L., Shi, W., Cunningham, A., and Huszar, F. (2017), "Lossy image compression with compressive autoencoders," in *International Conference on Learning Representations (ICLR)*.

Toderici, G., O'Malley, S. M., Hwang, S. J., Vincent, D., Minnen, D., Baluja, S., Covell, M., and Sukthankar, R. (2015), "Variable rate image compression with recurrent neural networks," *International Conference on Learning Representations (ICLR)*.

Toderici, G., Vincent, D., Johnston, N., Hwang, S. J., Minnen, D., Shor, J., and Covell, M. (2017), "Full resolution image compression with recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5435–5443.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. (2018), "Wasserstein auto-encoders," in *International Conference on Learning Representations (ICLR)*.

Torfason, R., Mentzer, F., Agustsson, E., Tschannen, M., Timofte, R., and Van Gool, L. (2018), "Towards image understanding from deep compression without decoding," in *International Conference on Learning Representations (ICLR)*.

Tschannen, M. (2014), *Dimensionality reduction for sparse subspace clustering*, Master's thesis, ETH Zürich.

Tschannen, M., Agustsson, E., and Lucic, M. (2018a), "Deep generative models for distribution-preserving lossy compression," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5933–5944.

Tschannen, M. and Bölcskei, H. (2015), "Nonparametric nearest neighbor random process clustering," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, pp. 1207–1211.

——— (2017), "Robust nonparametric nearest neighbor random process clustering," *IEEE Transactions on Signal Processing*, vol. 65, no. 22, pp. 6009–6023.

——— (2018), "Noisy subspace clustering via matching pursuits," *IEEE Transactions on Information Theory*, vol. 64, no. 6, pp. 4081–4104.

Tschannen, M., Khanna, A., and Anandkumar, A. (2018b), "StrassenNets: Deep learning with a multiplication budget," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 4992–5001.

Tucci, M. and Raugi, M. (2011), "Analysis of spectral clustering algorithms for linear and nonlinear time series," in *Proceedings of the IEEE Inter-*

national Conference on Intelligent Systems Design and Applications (ISDA), pp. 925–930.

Vershynin, R. (2012), “Non-asymptotic random matrix theory,” in Y. C. Eldar and G. Kutyniok (Eds.), *Compressed Sensing: Theory and Applications*, Cambridge University Press.

Vidal, R. (2011), “Subspace clustering,” *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68.

Vidal, R., Soatto, S., Ma, Y., and Sastry, S. (2003), “An algebraic geometric approach to the identification of a class of linear hybrid systems,” in *Proceeding of the IEEE Conference on Decision and Control*, pp. 167–172.

Vilar, J. A. and Pértega, S. (2004), “Discriminant and cluster analysis for Gaussian stationary processes: Local linear fitting approach,” *Journal of Nonparametric Statistics*, vol. 16, no. 3-4, pp. 443–462.

Villani, C. (2008), *Optimal transport: Old and new*, vol. 338, Springer Science & Business Media.

Vitaladevuni, S. N., Natarajan, P., Prasad, R., and Natarajan, P. (2011), “Efficient orthogonal matching pursuit using sparse random projections for scene and video classification,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2312–2319.

Wang, M., Liu, B., and Foroosh, H. (2016a), “Design of efficient convolutional layers using single intra-channel convolution, topological subdivisioning and spatial “bottleneck” structure,” *arXiv:1608.04337*.

Wang, Y., Liu, Z., and Huang, J.-C. (2000), “Multimedia content analysis – Using both audio and visual clues,” *IEEE Signal Processing Magazine*, vol. 17, no. 6, pp. 12–36.

Wang, Y., Wang, Y.-X., and Singh, A. (2016b), “Graph connectivity in noisy sparse subspace clustering,” in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 538–546.

Wang, Y.-X. and Xu, H. (2016), “Noisy sparse subspace clustering,” *Journal of Machine Learning Research*, vol. 17, no. 12, pp. 1–41.

Wen, W., Wu, C., Wang, Y., Chen, Y., and Li, H. (2016), “Learning structured sparsity in deep neural networks,” in *Advances in Neural Information Processing Systems (NIPS)*, pp. 2074–2082.

Wen, W., Xu, C., Wu, C., Wang, Y., Chen, Y., and Li, H. (2017), “Coordinating filters for faster deep neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 658–666.

White, H. (2014), *Asymptotic Theory for Econometricians*, Academic Press.

Wu, C.-Y., Singhal, N., and Krähenbühl, P. (2018), “Video compression through image interpolation," in *European Conference on Computer Vision (ECCV).*

Xiong, Y. and Yeung, D.-Y. (2004), "Time series clustering with ARMA mixtures," *Pattern Recognition*, vol. 37, no. 8, pp. 1675–1689.

Yan, D., Huang, L., and Jordan, M. I. (2009), "Fast approximate spectral clustering," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 907–916.

Yang, T.-J., Chen, Y.-H., and Sze, V. (2017), "Designing energy-efficient convolutional neural networks using energy-aware pruning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5687–5695.

You, C., Robinson, D., and Vidal, R. (2016), "Scalable sparse subspace clustering by orthogonal matching pursuit," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3918–3927.

Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., and Xiao, J. (2015), "LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv:1506.03365.*

Zhang, X., Zhou, X., Lin, M., and Sun, J. (2017), "Shufflenet: An extremely efficient convolutional neural network for mobile devices," *arXiv:1707.01083.*

Zhang, X., Zou, J., He, K., and Sun, J. (2016), "Accelerating very deep convolutional networks for classification and detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 1943–1955.

Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., and Zou, Y. (2016), "DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients," *arXiv:1606.06160.*

Zhu, C., Han, S., Mao, H., and Dally, W. J. (2016), "Trained ternary quantization," in *International Conference on Learning Representations (ICLR).*

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017a), "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2223–2232.

Zhu, J.-Y., Zhang, R., Pathak, D., Darrell, T., Efros, A. A., Wang, O., and Shechtman, E. (2017b), "Toward multimodal image-to-image translation," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 465–476.

Zhuang, B., Shen, C., Tan, M., Liu, L., and Reid, I. (2018), "Towards effective low-bitwidth convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7920–7928.