UNIVERSITY OF CALIFORNIA SAN DIEGO

Quantization for High-dimensional Data and Neural Networks: Theory and Algorithms

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Mathematics

by

Jinjie Zhang

Committee in charge:

Professor Rayan Saab, Chair
Professor Alexander Cloninger, Co-Chair
Professor Ery Arias-Castro
Professor Sanjoy Dasgupta

2023

The Dissertation of Jinjie Zhang is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2023

DEDICATION

To my wife, my parents, and people who helped me over the past five years.

EPIGRAPH

Number rules the universe.

—*Pythagoras*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Professor Rayan Saab who inspires me throughout my doctoral studies and helps me grow as a researcher with curiosity, self-reliance, and independent thinking. I enjoy and cherish the eureka moment when brainstorming new ideas with him in front of a white board. In addition, I would like to thank my co-advisor Professor Alexander Cloninger who gives me invaluable instruction in machine learning research and helps me to set up connections with other researchers. This dissertation would not have been possible without their immeasurable support and encouragement.

I would like to thank Professor Ery Arias-Castro, Professor Sanjoy Dasgupta, and Professor Kamalika Chaudhuri for serving as my committee members, offering meaningful discussions, and sharing invaluable career advice. I would also like to thank Professor Caroline Moosmüller, Professor Lek-Heng Lim, and Professor Shmuel Friedland for their help to crack challenging problems together. I appreciate all the research opportunities and resources provided by Amazon, AMD, and Lenovo, where I obtained industry experience from many excellent researchers: Samir Touzani, Jiangchuan Huang, Yanan Yu, Jonathan Toner, Michael Wagner, Alireza Khodamoradi, Zhicheng Fu.

Finally, I would like to express my deep gratitude to my wife Yichao (Rachel) and my parents Yumin and Ting. Without their tremendous understanding and encouragement in the past five years, it would be impossible for me to complete my doctoral journey.

Chapter 2, in full, is joint work with Rayan Saab and has been published in International Conference on Learning Representations (ICLR), 2021. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in full, is joint work with Harish Kannan, Alexander Cloninger, Rayan Saab, and has been submitted for publication. The dissertation author was the primary investigator and author of this paper.

Chapter 4, in full, is joint work with Yixuan Zhou, Rayan Saab, and has been published in the SIAM Journal on Mathematics of Data Science (SIMODS), 2023. The dissertation author

was the primary investigator and author of this material.

Chapter 5, in full, is joint work with Rayan Saab and is currently being prepared for submission for publication. The dissertation author was the primary investigator and author of this material.

VITA

2012–2016 Bachelor of Science in Information and Computing Science, Beijing Jiaotong University

2016–2018 Master of Science in Statistics, University of Chicago

2018–2023 Doctor of Philosophy in Mathematics, University of California San Diego

PUBLICATIONS

J. Zhang, Y. Zhou, R. Saab. "Post-training quantization for neural networks with provable guarantees". *SIAM Journal on Mathematics of Data Science* 5 (2), 373-399, 2023.

J. Zhang, J. Huang, Y. Yu. "The Markov chain model for DP engagement". *Amazon Machine Learning Conference (AMLC)*, 2022.

J. Zhang, H. Kannan, A. Cloninger, R. Saab, "Sigma-delta and distributed noise-shaping quantization methods for random Fourier features". *arXiv preprint arXiv:2106.02614*, 2021.

J. Zhang, R. Saab. "Faster binary embeddings for preserving euclidean distances". *International Conference on Learning Representations (ICLR)*, 2021.

S. Friedland, L. Lim, J. Zhang. "Grothendieck constant is norm of Strassen matrix multiplication tensor", *Numerische Mathematik* 143, 2019.

S. Friedland, L. Lim, J. Zhang. "An elementary and unified proof of Grothendieck's inequality", *L'Enseignement Mathématique* 64.3, 2019.

J. Zhang, S. Zheng. "On refined Hardy-Knopp type inequalities in Orlicz spaces and some related results". *Journal of Inequalities and Applications*, 2015.

ABSTRACT OF THE DISSERTATION

Quantization for High-dimensional Data and Neural Networks: Theory and Algorithms

by

Jinjie Zhang

Doctor of Philosophy in Mathematics

University of California San Diego, 2023

Professor Rayan Saab, Chair
Professor Alexander Cloninger, Co-Chair

Over the past few years, quantization has shown great and consistent success in compressing high-dimensional data and over-parameterized models. This dissertation focuses on theoretical guarantees and applications of quantization algorithms for fast binary embeddings (FBEs), random Fourier features (RFFs), and neural networks (NNs). Chapter 1 presents an introduction to quantization and background information for topics covered by later chapters.

In Chapter 2, we introduce a novel fast binary embedding algorithm that transforms data points from high-dimensional space into low-dimensional binary sequences. We prove that the ℓ_2 distances among original data points can be recovered by the ℓ_1 norm on binary embeddings and its associated approximation error is comparable to that of a continuous valued Johnson-Lindenstrauss embedding plus a quantization error that admits a polynomial decay as

the embedding dimension increases. So the length of the binary codes required to achieve a desired accuracy is quite small which is empirically verified by our experiments over natural images.

As a natural extension of Chapter 2, Chapter 3 proposes low-bit $\Sigma\Delta$ and distributed noise-shaping methods for quantizing RFFs associated with shift-invariant kernels. We show that the quantized RFFs achieve a high accuracy approximation of the underlying kernels with the approximation error decaying polynomially as the dimension of the RFFs increases, and decaying exponentially as a function of the bits used. Moreover, we test our method on multiple machine learning tasks that involve the kernel method.

In Chapter 4, we generalize a post-training neural-network quantization method, GPFQ, that is based on a greedy path-following mechanism. We expands the results of previous work on GPFQ to handle general quantization alphabets and a range of input distributions, showing that for quantizing a single-layer network, the relative square error essentially decays linearly in the number of weights – i.e., level of over-parametrization. Additionally, we propose modifications to promote sparsity of the weights, and rigorously analyze the associated error. Without fine-tuning, we can quantize several common architectures to 4 bits, while attaining an accuracy loss less than 1%.

Since the theoretical results in Chapter 4 are limited to single-layer neural networks, in Chapter 5, we propose a new stochastic algorithm for quantizing pretrained neural networks. We establish, for the first time, rigorous full-network error bounds, under an infinite alphabet condition and minimal assumptions on the weights and input data. Moreover, we demonstrate that it is possible to achieve error bounds equivalent to those obtained in the infinite alphabet case, using a mere $\log_2 \log N$ bits, where N represents the maximum width across all layers.

Chapter 1

Introduction

Quantization is a widely used process with applications in various fields, including signal acquisition and processing, as well as data and model compression. Its primary objective is to efficiently represent signals or data, usually by mapping a range of values to a smaller set of discrete values known as the quantization alphabet. When the quantization alphabet is finite, its elements can be enumerated hence easily represented using finite bit-strings. As a result, quantizing data in signal processing, or model parameters in machine learning offers potential benefits such as memory savings and simplified operations for faster computation. This thesis focuses on studying quantization algorithms and theory in the following contexts.

1. Geometry preserving data quantization: This involves mapping high-dimensional data to a low-dimensional discrete space while preserving important geometric information, such as Euclidean or kernel distances.

2. Model quantization: This refers to converting the parameters of deep neural networks from, say, 32-bit representations to lower-bit representations, while ensuring the network's performance is maintained.

Quantization enables information compression by reducing redundancies in high dimensional data and over-parameterized neural networks. While it offers memory and time efficiency, quantization also inevitably introduces an unavoidable approximation error that can negatively affect performance. Therefore, this thesis introduces and studies algorithms with favorable

trade-offs between controlling the quantization error and minimizing the number of bits used for encoding. Moreover, we provide rigorous error bounds for different families of quantization designs and compare their performance over various machine learning tasks. In this opening chapter, we review the mathematical foundations of quantization and introduce the topics covered by this thesis.

1.1 Quantization Fundamentals

Throughout this thesis, we focus on uniform symmetric quantization as it enables faster computation in the integer domain and allows high throughput in hardware. Specifically, it converts real and floating-point numbers to elements from fixed grid-like sets, called *alphabets*. In this thesis, an important alphabet we consider is symmetric with evenly distributed elements, given by

$$\mathscr{A} := \{\pm k\delta : 0 \leq k \leq K, k \in \mathbb{Z}\}. \tag{1.1}$$

Here $\delta > 0$ denotes the *step size* and we refer to $K \in \mathbb{N}$ as the *number of levels* of the alphabet. Note that (1.1) is widely used for neural network quantization which we study in Chapter 4 and Chapter 5. In many applications, including those considered in Chapter 2 and Chapter 3, one may also use a variant of (1.1) that excludes 0. Our discussion of quantization herein is motivated by this thesis's focus on applications in machine learning. Thus, we will next discuss computation and calibration issues that are salient to such applications. Then, we will introduce Sigma-Delta quantization, a prominent approach used in signal processing applications as it will be heavily used in Chapter 2 and Chapter 3.

1.1.1 Scalar Quantization

Scalar quantization is a prevalent method to perform quantization by leveraging the "rounding-to-nearest element" operation. For example, for the alphabet given by (1.1), we define

the associated *scalar quantizer* $Q : \mathbb{R} \to \mathscr{A}$ by

$$Q(z) := \arg\min_{p \in \mathscr{A}} |z - p| = \delta \operatorname{sign}(z) \min\left\{ \left| \left\lfloor \frac{z}{\delta} + \frac{1}{2} \right\rfloor \right|, K \right\}. \tag{1.2}$$

For a vector $y \in \mathbb{R}^m$, we generalize the scalar quantization by applying Q pointwise, that is, $Q(y) := (Q(y_1), Q(y_2), \ldots, Q(y_m)) \in \mathscr{A}^m$. Note that this operation is also called a *memoryless scalar quantizer* (MSQ) in later chapters.

1.1.2 Accelerated Computing

In practice, scalar quantization (1.2) can be decomposed into two steps:

(1) Convert a real or floating-point number z to an integer representation that is used for accelerated computing in hardware.

$$z_q := \operatorname{clip}\big(\operatorname{round}\big(\frac{z}{\delta}\big), K\big) \in \mathbb{Z}.$$

Here, for any $x \in \mathbb{R}$,

$$\operatorname{round}(x) := \arg\min_{i \in \mathbb{Z}} |x - i| \quad \text{and} \quad \operatorname{clip}(x, K) = \begin{cases} x & \text{if } |x| \le K, \\ -K & \text{if } x < -K, \\ K & \text{if } x > K. \end{cases}$$

(2) Transform z_q into a floating-point number $\hat{z}$:

$$\hat{z} := \delta z_q.$$

It is easy to verify that $\hat{z} = Q(z)$. So we use $Q(z)$ for theoretical analysis in most cases. However, to understand the impact of using quantization on computational cost, we will use the two-step process introduced above and take neural networks as an example. Consider a fully-connected

layer that performs a matrix multiplication $Y = XW$, where $X \in \mathbb{R}^{m\times p}$ is the input activation coming from previous layers, $W \in \mathbb{R}^{p\times n}$ is the weight matrix with n neurons (i.e. columns), and $Y \in \mathbb{R}^{m\times n}$ is the pre-activation for the current layer. Suppose that we quantize X and W pointwise with step size $\delta_a > 0$ and $\delta_w > 0$ respectively. Then the real-valued matrix multiplication can be approximated as follows

$$Y_{ij} = \sum_{k=1}^{p} X_{ik}W_{kj} \approx \sum_{k=1}^{p} \widehat{X}_{ik}\widehat{W}_{kj} = \delta_a\delta_w \sum_{k=1}^{p} (X_{ik})_q (W_{kj})_q.$$

Since $(X_{ik})_q, (W_{kj})_q \in \mathbb{Z}$, the approximation above invokes faster integer mutiply-add operations $\sum_{k=1}^{p}(X_{ik})_q(W_{kj})_q$ followed by a singe floating-point operation which is cheap. In general, rather than adopt a 32-bit floating point format for the model parameters, one uses significantly fewer bits for representing weights, activations, and even gradients. Since the floating-point operations are substituted by more efficient low-bit operations, quantization can reduce inference time and power consumption.

1.1.3 Calibration

The scalar quantizer Q in (1.2) has two parameters K and δ. For a specific quantization problem, the number of bits $b \in \mathbb{N}$ is usually given and thus $K = 2^{b-1}$ is fixed. It remains to determine the value of δ in the hope that the quantization error is minimized for multiple inputs in a batch. The process of choosing a proper step size $\delta > 0$ is called *calibration*. There are three commonly used calibration approaches:

(1) Max calibration: pick $\delta > 0$ such that the largest element $q_{\max} := K\delta \geq |z|$ for all values z seen during calibration, or simply for all values z in the input set of interest.

(2) Percentile calibration: set $\delta > 0$ such that $q_{\max}$ is a percentile of the distribution of absolute values $|z|$ seen during calibration.

(3) Entropy calibration: by selecting an optimal $\delta > 0$, the Kullback–Leibler (KL) divergence between the original floating-point distribution and the quantized distribution is minimized.

Apart from these calibration methods, one can learn the step size δ using backpropagation, see [41, 13].

1.1.4 Sigma-Delta Quantization

Although the MSQ in (1.2) can encode the input x with minimal computational cost, it does not adapt to the down-stream tasks for which quantization is being performed. For example, in various contexts, one is interested in minimizing $\|D(Q(x)) - D(x)\|$ rather than $\|Q(x) - x\|$, where D is an operator acting on the quantized data. In this case, simply minimizing the scalar error associated with each entry of the vector is far from optimal.

In contrast, *noise-shaping quantization* (see, e.g., [6, 7, 8, 1]) schemes attempt to push the quantization error $\mathrm{err}(x) := Q(x) - x$ as close to the kernel of the operator D as possible. In particular, in the case of linear D, one seeks $\|D(Q(x)) - D(x)\| = \|D(\mathrm{err}(x))\| \approx 0$. While placing the reconstruction error completely in the null space of the reconstruction operator is generally impossible, there are quantization schemes like $\Sigma\Delta$ *quantization* that utilizes quantization errors from previous steps to increase the overall accuracy of the quantized sequence. As a result, $\Sigma\Delta$ quantization offers much better error guarantees than MSQ in a variety of applications, ranging from bandlimited function quantization [10, 17, 11, 4], to quantization of compressed sensing measurements [22, 33, 32, 29], and as we will see in Chapter 2 and Chapter 3, to the quantization of high-dimensional data [14, 37] and random Fourier features [31].

Given $r \in \mathbb{N}$, an r-th order $\Sigma\Delta$ quantizer $Q^{(r)} : \mathbb{R}^m \to \mathscr{A}^m$ maps an input signal $y = (y_i)_{i=1}^m \in \mathbb{R}^m$ to a quantized sequence $q = (q_i)_{i=1}^m \in \mathscr{A}^m$ via a quantization rule ρ and the following iterations. For $i = 1, 2, \ldots, m$,

$$\begin{cases} u_0 = u_{-1} = \ldots = u_{1-r} = 0, \\ q_i = Q(\rho(y_i, u_{i-1}, \ldots, u_{i-r})), \\ u_i = \sum_{j=1}^r (-1)^{j-1} \binom{r}{j} u_{i-j} + y_i - q_i, \end{cases} \tag{1.3}$$

where $Q(z)$ is the scalar quantizer as in (1.2) and $u \in \mathbb{R}^m$ is the *state* vector. Note that the last equation in (1.3) is equivalent to

$$D^r u = y - q \tag{1.4}$$

where $D \in \mathbb{R}^{m\times m}$ is the first order difference matrix defined by

$$D_{ij} = \begin{cases} 1 & \text{if } i = j, \\ -1 & \text{if } i = j+1, \\ 0 & \text{otherwise.} \end{cases}$$

An important example is the first order $\Sigma\Delta$ quantizer in which we choose $r = 1$ and quantization rule $\rho(y_i, u_{i-1}, \ldots, u_{i-r}) = y_i + u_{i-1}$. It follows that $Q^{(1)}$ is given by

$$\begin{cases} u_0 = 0, \\ q_i = Q(y_i + u_{i-1}), \\ u_i = u_{i-1} + y_i - q_i. \end{cases} \tag{1.5}$$

Moreover, the quantization scheme $Q^{(r)}$ in (1.3) is *stable* if there exists $\mu > 0$ such that for each input with $\|y\|_\infty \leq \mu$, the state vector $u \in \mathbb{R}^m$ satisfies $\|u\|_\infty \leq C$. Here, μ and C are constants that do not depend on m. Stability heavily depends on the choice of quantization rule ρ and it is essential for controlling the quantization error $y - q = D^r u$ via boundedness. Although (1.5) is proved to be stable [11], it is a non-trivial task to design a stable $Q^{(r)}$ for $r > 1$. To achieve this goal, we adopt the techniques used in [11]. Specifically, an r-th order $\Sigma\Delta$ quantization scheme can arise from the following difference equation

$$y - q = H * v \tag{1.6}$$

where $*$ is the convolution operator and the sequence $H := D^r g$ with $g \in \ell^1$. Then any bounded

solution v of (1.6) gives rise to a bounded solution u of (1.4) via $u = g * v$. By change of variables, (1.4) can be reformulated as (1.6). By choosing a proper filter $h := \delta^{(0)} - H$, where $\delta^{(0)}$ denotes the Kronecker delta sequence supported at 0, one can implement (1.6) by $v_i = (h*v)_i + y_i - q_i$ and the corresponding stable quantization scheme $Q^{(r)}$ reads as

$$\begin{cases} q_i = Q((h*v)_i + y_i), \\ v_i = (h*v)_i + y_i - q_i. \end{cases} \tag{1.7}$$

Proposition 1.1.1 ([11, 21])**.** *There exists a universal constant $C > 0$ such that the $\Sigma\Delta$ schemes* (1.5) *and* (1.7) *are stable, and*

$$\|y\|_\infty \le \mu < 1 \Longrightarrow \|u\|_\infty \le c(K,r) := \frac{CC_1^r r^r}{2K-1},$$

where $C_1 = \left(\left\lceil \frac{\pi^2}{(\cosh^{-1}\gamma)^2} \right\rceil \frac{e}{\pi}\right)$ *with* $\gamma := 2K - (2K-1)\mu$.

1.2 Thesis Structure

Throughout this dissertation, we will focus on the scalar quantizer (1.2), the stable noise-shaping quantizer (1.3), and their applications for data quantization in Chapter 2 and Chapter 3, and model compression in Chapter 4 and Chapter 5. These topics are briefly introduced below and we will take a deep dive into each of them in later chapters.

1.2.1 Data Quantization.

Large-scale high-dimensional data has become increasingly common nowadays, which challenges machine learning algorithms to extract and preserve discriminative information from the data. As an important branch of representation learning [2, 40, 20], fast binary embedding (FBE) [38, 16, 39, 24, 44] methods quantize high-dimensional data into binary sequences such that input distances can be recovered from the binary codes. So we can perform efficient learning and similarity search, e.g. for image retrieval, directly in the binary space. In Chapter 2, we

propose a new FBE method to preserve pairwise ℓ_2 distances and rigorously analyze the trade-off between the approximation accuracy and the embedding dimension.

Although Kernel methods [34, 35] have long been demonstrated as effective techniques in various machine learning tasks such as support vector machines, logistic regression, and dimensionality reduction, they have limited scalability for large datasets. Specifically, an $N \times N$ kernel matrix derived from N data points suffer from $O(N^2)$ storage cost and $O(N^3)$ computational cost for common learning tasks. In order to overcome this bottleneck, one popular approach is to "linearize" the kernel by using random Fourier features (RFFs) [31]. Moreover, a low-precision quantization of RFFs [42, 43, 25] can further speed up training and alleviate the memory burden for large-scale data. Chapter 3 applies $\Sigma\Delta$ and distributed noise-shaping methods for quantizing the RFFs with low bitwidth and shows an excellent trade-off between memory use and accuracy.

1.2.2 Model Quantization.

The past decade has witnessed the resurrection of deep learning in many tasks, such as computer vision (CV), natural language processing (NLP), and multimodal learning, among others. Nevertheless, over-parameterized deep neural networks (DNNs) are computationally expensive to train, memory intensive to store, and energy consuming to apply. For example, pretrained generative models from the transformer [36] family, commonly known as GPT or OPT [30, 3, 46], have achieved great success in various applications, including zero-shot and few-shot learning.

Released in 2020, GPT-3 [3] marked a significant milestone in generative AI. With an astounding 175 billion parameters, it requires 800 GB to store and incurs a training cost of over 4.6 million dollars. The example of GPT-3 demonstrates the importance of model compression while maintaining performance, making this an important active area of deep learning research [18, 12, 15]. As a prominent approach to compress DNNs, quantization uses significantly fewer bits to represent weights and activations, which reduces inference time and

power consumption. Following [23], we can categorize neural network quantization methods into two classes: *quantization-aware training* (QAT) [13, 5, 9, 19] and *post-training quantization* (PTQ) [26, 27, 45, 28]. QAT retrains the quantized model and requires the training dataset to perform end-to-end backpropagation. In contrast to QAT, PTQ directly quantizes pretrained DNNs without retraining and it only needs a small amount of data.

In Chapter 4, we study a PTQ method for quantizing the weights of pretrained DNNs called *greedy path following quantization* (GPFQ). Moreover, we substantially improve GPFQ's theoretical analysis, propose a modification to handle convolutional layers, and propose a sparsity promoting version to encourage the algorithm to set many of the weights to zero. We demonstrate that the performance of our quantization methods is not only good in experimental settings, but, equally importantly, has favorable and rigorous error guarantees. Note that all technical proofs in Chapter 4 only apply for a single-layer neural network with certain assumed input distributions. This limitation naturally comes from the fact that a random input and a deterministic quantizer lead to intractable distributions after passing through multiple nonlinear layers.

To overcome this main obstacle to obtaining theoretical guarantees for multiple layer neural networks, in Chapter 5, we propose a new stochastic quantization framework, called stochastic path following quantization (SGPFQ), which introduces randomness into the quantizer. For the first time, we prove rigorous quantization error bounds for multi-layer neural networks, under both infinite and finite alphabet conditions. Moreover, by quantizing several common neural network architectures, we empirically show that the developed method presents only minor loss of accuracy compared to unquantized models.

References

[1] John J Benedetto, Alexander M Powell, and Ozgur Yilmaz. "Sigma-delta quantization and finite frames". In: *IEEE Transactions on Information Theory* 52.5 (2006), pp. 1990–2005.

[2] Yoshua Bengio, Aaron Courville, and Pascal Vincent. “Representation learning: A review and new perspectives”. In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. “Language models are few-shot learners”. In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[4] Emmanuel J Candès, Justin Romberg, and Terence Tao. “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information”. In: *IEEE Transactions on information theory* 52.2 (2006), pp. 489–509.

[5] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. “Pact: Parameterized clipping activation for quantized neural networks”. In: *arXiv preprint arXiv:1805.06085* (2018).

[6] Evan Chou and C Sinan Güntürk. “Distributed noise-shaping quantization: I. Beta duals of finite frames and near-optimal quantization of random measurements”. In: *Constructive Approximation* 44.1 (2016), pp. 1–22.

[7] Evan Chou and C Sinan Güntürk. “Distributed noise-shaping quantization: II. Classical frames”. In: *Excursions in Harmonic Analysis, Volume 5*. Springer, 2017, pp. 179–198.

[8] Evan Chou, C Sinan Güntürk, Felix Krahmer, Rayan Saab, and Özgür Yılmaz. “Noise-shaping quantization methods for frame-based and compressive sampling systems”. In: *Sampling theory, a renaissance* (2015), pp. 157–184.

[9] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. “Binaryconnect: Training deep neural networks with binary weights during propagations”. In: *Advances in neural information processing systems*. 2015, pp. 3123–3131.

[10] Ingrid Daubechies and Ron DeVore. “Approximating a bandlimited function using very coarsely quantized data: A family of stable sigma-delta modulators of arbitrary order”. In: *Annals of mathematics* (2003), pp. 679–710.

[11] Percy Deift, Felix Krahmer, and C Sınan Güntürk. “An optimal family of exponentially accurate one-bit Sigma-Delta quantization schemes”. In: *Communications on Pure and Applied Mathematics* 64.7 (2011), pp. 883–919.

[12] Lei Deng, Guoqi Li, Song Han, Luping Shi, and Yuan Xie. “Model compression and hardware acceleration for neural networks: A comprehensive survey”. In: *Proceedings of the IEEE* 108.4 (2020), pp. 485–532.

[13] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. “Learned step size Quantization”. In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=rkgO66VKDS.

[14] Simon Foucart and Holger Rauhut. “An invitation to compressive sensing”. In: *A mathematical introduction to compressive sensing*. Springer, 2013, pp. 1–39.

[15] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. “A survey of quantization methods for efficient neural network inference”. In: *arXiv preprint arXiv:2103.13630* (2021).

[16] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. “Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval”. In: *IEEE transactions on pattern analysis and machine intelligence* 35.12 (2012), pp. 2916–2929.

[17] C Sinan Güntürk. “One-bit sigma-delta quantization with exponential accuracy”. In: *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 56.11 (2003), pp. 1608–1630.

[18] Yunhui Guo. "A survey on methods and theories of quantized neural networks". In: *arXiv preprint arXiv:1808.04752* (2018).

[19] Song Han, Huizi Mao, and William J Dally. "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding". In: *arXiv preprint arXiv:1510.00149* (2015).

[20] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. "Contrastive representation learning: A framework and review". In: *Ieee Access* 8 (2020), pp. 193907–193934.

[21] Felix Krahmer, Rayan Saab, and Rachel Ward. "Root-exponential accuracy for coarse quantization of finite frame expansions". In: *IEEE transactions on information theory* 58.2 (2012), pp. 1069–1079.

[22] Felix Krahmer, Rayan Saab, and Özgür Yilmaz. "Sigma–delta quantization of sub-gaussian frame expansions and its application to compressed sensing". In: *Information and Inference: A Journal of the IMA* 3.1 (2014), pp. 40–58.

[23] Raghuraman Krishnamoorthi. "Quantizing deep convolutional networks for efficient inference: A whitepaper". In: *arXiv preprint arXiv:1806.08342* (2018).

[24] Ping Li, Anshumali Shrivastava, Joshua L Moore, and Arnd C König. "Hashing algorithms for large-scale learning". In: *Advances in neural information processing systems*. 2011, pp. 2672–2680.

[25] Xiaoyun Li and Ping Li. "Quantization algorithms for random fourier features". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 6369–6380.

[26] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. "Brecq: Pushing the limit of post-training quantization by block reconstruction". In: *International Conference on Learning Representations* (2021).

[27] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen B. "Up or down? adaptive rounding for post-training quantization". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 7197–7206.

[28] Yury Nahshan, Brian Chmiel, Chaim Baskin, Evgenii Zheltonozhskii, Ron Banner, Alex M Bronstein, and Avi Mendelson. "Loss aware post-training quantization". In: *Machine Learning* 110.11-12 (2021), pp. 3245–3262.

[29] Alexander M Powell, Rayan Saab, and Özgür Yılmaz. "Quantization and finite frames". In: *Finite Frames: Theory and Applications* (2013), pp. 267–302.

[30] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[31] Ali Rahimi and Benjamin Recht. "Random features for large-scale kernel machines". In: *Advances in neural information processing systems* 20 (2007), pp. 1177–1184.

[32] Rayan Saab, Rongrong Wang, and Özgür Yılmaz. "From compressed sensing to compressed bit-streams: practical encoders, tractable decoders". In: *IEEE Transactions on Information Theory* 64.9 (2017), pp. 6098–6114.

[33] Rayan Saab, Rongrong Wang, and Özgür Yılmaz. "Quantization of compressive samples with stable and robust recovery". In: *Applied and Computational Harmonic Analysis* 44.1 (2018), pp. 123–143.

[34] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Adaptive Computation and Machine Learning series, 2018.

[35] John Shawe-Taylor, Nello Cristianini, et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[37] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*. Vol. 47. Cambridge university press, 2018.

[38] Xinyang Yi, Constantine Caramanis, and Eric Price. "Binary embedding: Fundamental limits and fast algorithm". In: *International Conference on Machine Learning*. 2015, pp. 2162–2170.

[39] Felix Yu, Sanjiv Kumar, Yunchao Gong, and Shih-Fu Chang. "Circulant binary embedding". In: *International conference on machine learning*. 2014, pp. 946–954.

[40] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. "Network representation learning: A survey". In: *IEEE transactions on Big Data* 6.1 (2018), pp. 3–28.

[41] Dongqing Zhang, Jiaolong Yang, Dongqiangzi Ye, and Gang Hua. "Lq-nets: Learned quantization for highly accurate and compact deep neural networks". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 365–382.

[42] Jian Zhang, Avner May, Tri Dao, and Christopher Ré. "Low-precision random Fourier features for memory-constrained kernel approximation". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 1264–1274.

[43] Jinjie Zhang, Harish Kannan, Alexander Cloninger, and Rayan Saab. "Sigma-delta and distributed noise-shaping quantization methods for random fourier features". In: *arXiv preprint arXiv:2106.02614* (2021).

[44] Jinjie Zhang and Rayan Saab. "Faster Binary Embeddings for Preserving Euclidean Distances". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=YCXrx6rRCXO.

[45] Jinjie Zhang, Yixuan Zhou, and Rayan Saab. “Post-training quantization for neural networks with provable guarantees”. In: *SIAM Journal on Mathematics of Data Science* 5.2 (2023), pp. 373–399.

[46] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. “OPT: Open pre-trained transformer language models”. In: *arXiv preprint arXiv:2205.01068* (2022).

Chapter 2

Faster Binary Embeddings for Preserving Euclidean Distances

We propose a fast, distance-preserving, binary embedding algorithm to transform a high-dimensional dataset $\mathscr{T} \subseteq \mathbb{R}^n$ into binary sequences in the cube $\{\pm 1\}^m$. When $\mathscr{T}$ consists of well-spread (i.e., non-sparse) vectors, our embedding method applies a stable noise-shaping quantization scheme to Ax where $A \in \mathbb{R}^{m\times n}$ is a sparse Gaussian random matrix. This contrasts with most binary embedding methods, which usually use $x \mapsto \text{sign}(Ax)$ for the embedding. Moreover, we show that Euclidean distances among the elements of $\mathscr{T}$ are approximated by the ℓ_1 norm on the images of $\{\pm 1\}^m$ under a fast linear transformation. This again contrasts with standard methods, where the Hamming distance is used instead. Our method is both fast and memory efficient, with time complexity $O(m)$ and space complexity $O(m)$ on well-spread data. When the data is not well-spread, we show that the approach still works provided that data is transformed via a Walsh-Hadamard matrix, but now the cost is $O(n\log n)$ per data point. Further, we prove that the method is accurate and its associated error is comparable to that of a continuous valued Johnson-Lindenstrauss embedding plus a quantization error that admits a polynomial decay as the embedding dimension m increases. Thus the length of the binary codes required to achieve a desired accuracy is quite small, and we show it can even be compressed further without compromising the accuracy. To illustrate our results, we test the proposed method on natural images and show that it achieves strong performance.

2.1 Introduction

Analyzing large data sets of high-dimensional raw data is usually computationally demanding and memory intensive. As a result, it is often necessary as a preprocessing step to transform data into a lower-dimensional space while approximately preserving important geometric properties, such as pairwise ℓ_2 distances. As a critical result in dimensionality reduction, the Johnson-Lindenstrauss (JL) lemma [23] guarantees that every finite set $\mathscr{T} \subseteq \mathbb{R}^n$ can be (linearly) mapped to a $m = O(\varepsilon^{-2}\log(|\mathscr{T}|))$ dimensional space in such a way that all pairwise distances are preserved up to an ε-Lipschitz distortion. Additionally, there are many significant results to speed up the JL transform by introducing fast embeddings, e.g. [1, 2, 27, 32], or by using sparse matrices [25, 24, 6]. Such fast embeddings can usually be computed in $O(n\log n)$ versus the $O(mn)$ time complexity of JL transforms that rely on unstructured dense matrices.

2.1.1 Related Work

To further reduce memory requirements, progress has been made in *nonlinearly* embedding high-dimensional sets $\mathscr{T} \subseteq \mathbb{R}^n$ to the binary cube $\{-1,1\}^m$ with $m \ll n$, a process known as binary embedding. Provided that $d_1(\cdot,\cdot)$ is a metric on $\mathbb{R}^n$, a distace preserving binary embedding is a map $f : \mathscr{T} \to \{-1,1\}^m$ and a function $d_2(\cdot,\cdot)$ on $\{-1,1\}^m \times \{-1,1\}^m$ to approximate distances, i.e.,

$$|d_2(f(x),f(y)) - d_1(x,y)| \leq \alpha, \quad \text{for } \forall x,y \in \mathscr{T}. \tag{2.1}$$

The potential dimensionality reduction ($m \ll n$) and 1-bit representation per dimension imply that storage space can be considerably reduced and downstream applications like learning and retrieval can happen directly using bitwise operations. Most existing nonlinear mappings f in (2.1) are generated using simple memory-less scalar quantization (MSQ). For example, given a set of unit vectors $\mathscr{T} \subseteq \mathbb{S}^{n-1}$ with finite size $|\mathscr{T}|$, consider the map

$$q_x := f(x) = \text{sign}(Gx) \tag{2.2}$$

where $G \in \mathbb{R}^{m \times n}$ is a standard Gaussian random matrix and $\text{sign}(\cdot)$ returns the element-wise sign of its argument. Let $d_1(x,y) = \frac{1}{\pi}\arccos(\|x\|_2^{-1}\|y\|_2^{-1}\langle x,y\rangle)$ be the normalized angular distance and $d_2(q_x,q_y) = \frac{1}{2m}\|q_x - q_y\|_1$ be the normalized Hamming distance. Then, Yi, Caramanis, and Price [40] show that (2.1) holds with probability at least $1-\eta$ if $m \gtrsim \alpha^{-2}\log(|\mathscr{T}|/\eta)$, so one can approximate geodesic distances with normalized Hamming distances. While this approach achieves optimal bit complexity (up to constants) [40], it has been observed in practice that m is usually around $O(n)$ to guarantee reasonable accuracy [15, 37, 41]. Much like linear JL embedding techniques admit fast counterparts, fast binary embedding algorithms have been developed to significantly reduce the runtime of binary embeddings [17, 30, 16, 15, 29, 35]. Indeed, fast JL transforms (FJLT) and Gaussian Toeplitz matrices [40], structured hashed projections [4], iterative quantization [17], bilinear projection [15], circulant binary embedding [41, 13, 12, 33, 26], sparse projection [39], and fast orthogonal projection [42] have all been considered.

These methods can decrease time complexity to $O(n\log n)$ operations per embedding, but still suffer from some important drawbacks. Notably, due to the sign function, these algorithms completely discard all magnitude information, as $\text{sign}(Ax) = \text{sign}(A(\alpha x))$ for all $\alpha > 0$. So, all points in the same direction embed to the same binary vector and cannot be distinguished. Even if one settles for recovering geodesic distances, using the sign function in (2.2) is an instance of MSQ so the estimation error α in (2.1) decays slowly as the number of bits m increases [40].

In addition to the above data independent approaches, there are data dependent embedding methods for distance recovery, including product quantization [22, 14], LSH-based methods [3, 38, 7] and iterative quantization [18]. Their accuracy, which can be excellent, nevertheless depends on the underlying distribution of the input dataset. Moreover, they may be associated with larger time and space complexity for embedding the data. For example, product quantization performs k-means clustering in each subspace to find potential centroids and stores associated lookup tables. LSH-based methods need random shifts and dense random projections to quantize each input data point.

Recently Huynh and Saab [21] resolved these issues by replacing the simple sign function with a Sigma-Delta ($\Sigma\Delta$) quantization scheme, or alternatively other noise-shaping schemes (see [5]) whose properties will be discussed in Section 2.3. They use the binary embedding

$$q_x := Q(DBx) \tag{2.3}$$

where Q is now a stable $\Sigma\Delta$ quantization scheme, $D \in \mathbb{R}^{m\times m}$ is a diagonal matrix with random signs, and $B \in \mathbb{R}^{m\times n}$ are specific structured random matrices. To give an example of $\Sigma\Delta$ quantization in this context, consider $w := DBx$. Then the simplest $\Sigma\Delta$ scheme computes q_x via the following iteration, run for $i = 1, ..., m$:

$$\begin{cases} u_0 = 0, \\ q_x(i) = \text{sign}(w_i + u_{i-1}), \\ u_i = u_{i-1} + w_i - q_i. \end{cases} \tag{2.4}$$

The choices of B in [21] allow matrix vector multiplication to be implemented using the fast Fourier transform. Then the original Euclidean distance $\|x-y\|_2$ can be recovered via a pseudo-metric on the quantized vectors given by

$$d_{\widetilde{V}}(q_x, q_y) := \|\widetilde{V}(q_x - q_y)\|_2 \tag{2.5}$$

where $\widetilde{V} \in \mathbb{R}^{p\times m}$ is a "normalized condensation operator", a sparse matrix that can be applied fast (see Section 2.3). Regarding the complexity of applying (2.3) to a single $x \in \mathbb{R}^n$, note that $x \mapsto DBx$ has time complexity $O(n \log n)$ while the quantization map needs $O(m)$ time and results in an m bit representation. So when $m \leq n$, the total time complexity for (2.3) is around $O(n \log n)$.

2.1.2 Methods and Contributions

We extend these results by replacing DB in (2.3) by a sparse Gaussian matrix $A \in \mathbb{R}^{m\times n}$ so that now

$$q_x := Q(Ax). \tag{2.6}$$

Given scaled high-dimensional data $\mathscr{T} \subset \mathbb{R}^n$ contained in the ℓ_2 ball $B_2^n(\kappa)$ with radius κ, we put forward Algorithm 1 to generate binary sequences and Algorithm 2 to compute estimates of the Euclidean distances between elements of $\mathscr{T}$ via an ℓ_1-norm rather than ℓ_2-norm. The contribution of this work is threefold. First, we prove Theorem 2.1.1 quantifying the performance of our algorithms.

Algorithm 1: Fast Binary Embedding for Finite $\mathscr{T}$

Input: $\mathscr{T} = \{x^{(j)}\}_{j=1}^k \subseteq B_2^n(\kappa)$ ▷ Data points in ℓ_2 ball

Generate $A \in \mathbb{R}^{m\times n}$ as in Definition 2.2.2 ▷ Sparse Gaussian matrix A

for $j \leftarrow 1$ **to** k **do**

  $z^{(j)} \leftarrow Ax^{(j)}$

  $q^{(j)} = Q(z^{(j)})$ ▷ Stable $\Sigma\Delta$ quantizer Q as in (2.4), or more generally (2.21).

Output: Binary sequences $\mathscr{B} = \{q^{(j)}\}_{j=1}^k \subseteq \{-1,1\}^m$

Algorithm 2: ℓ_2 Norm Distance Recovery

Input: $q^{(i)}, q^{(j)} \in \mathscr{B}$ ▷ Binary sequences produced by Algorithm 1

$y^{(i)} \leftarrow \widetilde{V} q^{(i)}$ ▷ Condense the components of q

$y^{(j)} \leftarrow \widetilde{V} q^{(j)}$

Output: $\|y^{(i)} - y^{(j)}\|_1$ ▷ Approximation of $\|x^{(i)} - x^{(j)}\|_2$

Theorem 2.1.1 (Main result)**.** *Let $\mathscr{T} \subseteq \mathbb{R}^n$ be a finite, appropriately scaled set with elements satisfying $\|x\|_\infty = O(n^{-1/2}\|x\|_2)$ and $\|x\|_2 \le \kappa < 1$. If $m \gtrsim p := \Omega(\varepsilon^{-2}\log(|\mathscr{T}|^2/\delta))$ and $r \ge 1$ is the integer order of Q, then with probability $1-2\delta$ on the draw of the sparse Gaussian matrix A, the following holds uniformly over all x, y in $\mathscr{T}$: Embedding x, y into $\{-1,1\}^m$ using Algorithm 1, and estimating the associated distance between them using Algorithm 2 yields the*

error bound

$$\left| d_{\widetilde{V}}(q_x, q_y) - \|x-y\|_2 \right| \leq c \left(\frac{m}{p}\right)^{-r+1/2} + \varepsilon \|x-y\|_2$$

where $c > 0$ is a constant.

Theorem 2.1.1 yields an approximation error bounded by two components, one due to quantization and another that resembles the error from a *linear* JL embedding into a p-dimensional space. The latter part is essentially proportional to $p^{-1/2}$, while the quantization component decays polynomially fast in m, and can be made harmless by increasing m. Moreover, the number of bits $m \gtrsim \varepsilon^{-2} \log(|\mathscr{T}|)$ achieves the optimal bit complexity required by any oblivious random embedding that preserves Euclidean or squared Euclidean distance, see Theorem 4.1 in [11]. Theorem 2.4.2 is a more precise version of Theorem 2.1.1, with all quantifiers, and scaling parameters specified explicitly, and with a potential modification to A that enables the result to hold for arbitrary (not necessarily well-spread) finite $\mathscr{T}$, at the cost of increasing the computational complexity of embedding a point to $O(n \log n)$. We also note that if the data did not satisfy the scaling assumption of Theorems 2.1.1 and 2.4.2, then one can replace $\{-1, 1\}$ by $\{-C, C\}$, and the quantization error would scale by C.

Second, due to the sparsity of A, (2.6) can be computed much faster than (2.3), when restricting our results to "well-spread" vectors x, i.e., those that are not sparse. On the other hand, in Section 2.5, we show that Algorithm 1 achieves $O(m)$ time and space complexity in contrast with the common $O(n \log n)$ runtime of fast binary embeddings, e.g., [15, 40, 41, 13, 12, 21] that rely on fast JL transforms or circulant matrices. Meanwhile, Algorithm 2 requires only $O(m)$ runtime.

Third, Definition 2.2.3 shows that $\widetilde{V}$ is sparse and essentially populated by integers bounded by $(m/p)^r$ where r, m, p are as in Theorem 2.1.1. In Section 2.5, we note that each $y^{(i)} = \widetilde{V} q^{(i)}$ (and the distance query), can be represented by $O(p \log_2(m/p))$ bits, instead of m bits, without affecting the reconstruction accuracy. This is a consequence of using the ℓ_1-norm in Algorithm 2. Had we instead used an ℓ_2-norm, we would have required $O(p(\log_2(m/p))^2)$ bits.

Finally, we remark that while the assumption that the vectors x are well-spread (i.e. $\|x\|_\infty = O(n^{-1/2}\|x\|_2)$) may appear restrictive, there are important instances where it holds. Natural images seem to be one such case, as are random Fourier features [36]. Similarly, Gaussian (and other subgaussian) random vectors satisfy a slightly weakened $\|x\|_\infty = O(\log(n)n^{-1/2}\|x\|_2)$ assumption with high probability, and one can modify our construction by slightly reducing the sparsity of A (and slightly increasing the computational cost) to handle such vectors. On the other hand, if the data simply does not satisfy such an assumption, one can still apply Theorem 2.4.2 part (ii), but now the complexity of embedding a point is $O(n\log n)$.

2.2 Preliminaries

2.2.1 Notation and definitions

Throughout, $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$ mean that $|f(n)|$ is bounded above and below respectively by a positive function $g(n)$ up to constants asymptotically; that is, $\limsup_{n\to\infty} \frac{|f(n)|}{g(n)} < \infty$. Similarly, we use $f(n) = \Theta(g(n))$ to denote that $f(n)$ is bounded both above and below by a positive function $g(n)$ up to constants asymptotically. We next define operator norms.

Definition 2.2.1. *Let $\alpha, \beta \in [1,\infty]$ be integers. The (α,β) operator norm of $K \in \mathbb{R}^{m\times n}$ is* $\|K\|_{\alpha,\beta} = \max_{x\neq 0} \frac{\|Kx\|_\beta}{\|x\|_\alpha}$.

We now introduce some notation and definitions that are relevant to our construction.

Definition 2.2.2 (Sparse Gaussian random matrix)**.** *Let $A = (a_{ij}) \in \mathbb{R}^{m\times n}$ be a random matrix with i.i.d. entries such that a_{ij} is 0 with probability $1-s$ and is drawn from $\mathcal{N}(0, \frac{1}{s})$ with probability s.*

We adopt the definition of a condensation operator of Chou and Güntürk [5] and Huynh and Saab [21].

Definition 2.2.3 (Condensation operator)**.** *Let p, r, λ be fixed positive integers such that $\lambda = r\widetilde{\lambda} - r + 1$ for some integer $\widetilde{\lambda}$. Let $m = \lambda p$ and v be a row vector in $\mathbb{R}^{\lambda}$ whose entry v_j is the j-th coefficient of the polynomial $(1 + z + \ldots + z^{\widetilde{\lambda}-1})^r$. Define the condensation operator $V \in \mathbb{R}^{p \times m}$ by*

$$V = I_p \otimes v = \begin{bmatrix} v & & \\ & \ddots & \\ & & v \end{bmatrix}.$$

For example, when $r = 1$, $\lambda = \widetilde{\lambda}$, and $v \in \mathbb{R}^{\lambda}$ is simply the vector of all ones. The normalized condensation operator is given by

$$\widetilde{V} = \frac{\sqrt{\pi/2}}{p\|v\|_2} V.$$

The fast JL transform was first studied by Ailon and Chazelle [1]. It admits many variants and improvements, e.g. [27, 31]. The idea is that given any $x \in \mathbb{R}^n$ we use a fast "Fourier-like" transform, like the Walsh-Hadamard transform, to distribute the total mass (i.e. $||x||_2$) of x relatively evenly to its coordinates.

Definition 2.2.4 (FJLT)**.** *The fast JL transform can be obtained by*

$$\Phi := AHD \in \mathbb{R}^{m \times n}. \tag{2.7}$$

Here, $A \in \mathbb{R}^{m \times n}$ is a sparse Gaussian random matrix, as in Definition 2.2.2, while $H \in \mathbb{R}^{n \times n}$ is a normalized Walsh-Hadamard matrix defined by $H_{ij} = n^{-1/2}(-1)^{\langle i-1, j-1 \rangle}$ where $\langle i, j \rangle$ is the bitwise dot product of the binary representations of the numbers i and j. Finally, $D \in \mathbb{R}^{n \times n}$ is diagonal with diagonal entries drawn independently from $\{-1, 1\}$ with probability $1/2$ for each.

2.2.2 condensed Johnson-Lindenstrauss Transforms

Definition 2.2.5. *When $\widetilde{V}$ is a condensation operator, and A is a sparse Gaussian, we refer to $\widetilde{V}A$ as a condensed sparse JL transform (CSJLT). When A is replaced by Φ as in Definition 2.2.4 we refer to $\widetilde{V}\Phi$ as a condensed fast JL transform (CFJLT).*

The definition above is justified by the following lemma (see Appendix 2.8 for the proof).

Lemma 2.2.6 (CJLT lemma)**.** *Let $\mathscr{T}$ be a finite subset of $\mathbb{R}^n$, $\lambda \in \mathbb{N}$, $\varepsilon \in (0, \frac{1}{2})$, $\delta \in (0,1)$, $p = O(\varepsilon^{-2}\log(|\mathscr{T}|^2/\delta)) \in \mathbb{N}$ and $m = \lambda p$. Let $\widetilde{V} \in \mathbb{R}^{p\times m}$ be as in Definition 2.2.3, $A \in \mathbb{R}^{m\times n}$ be the sparse Gaussian matrix in Definition 2.2.2 with $s = \Theta(\varepsilon^{-1}n^{-1}(\|v\|_\infty/\|v\|_2)^2) \leq 1$, and $Phi = AHD \in \mathbb{R}^{m\times n}$ be the FJLT in Definition 2.2.4 with $s = \Theta(\varepsilon^{-1}n^{-1}(\|v\|_\infty/\|v\|_2)^2\log n) \leq 1$. If $\mathscr{T}$ consists of well-spread vectors, that is, $\|x\|_\infty = O(n^{-1/2}\|x\|_2)$ for all $x \in \mathscr{T}$, then*

$$\left| \|\widetilde{V}A(x-y)\|_1 - \|x-y\|_2 \right| \leq \varepsilon\|x-y\|_2 \tag{2.8}$$

holds uniformly for all $x, y \in \mathscr{T}$ with probability at least $1-\delta$. If $\mathscr{T}$ is finite but arbitrary, then

$$\left| \|\widetilde{V}Phi(x-y)\|_1 - \|x-y\|_2 \right| \leq \varepsilon\|x-y\|_2 \tag{2.9}$$

holds uniformly for all $x, y \in \mathscr{T}$ with probability at least $1-\delta$.

So $\mathscr{T} \subseteq \mathbb{R}^n$ is embedded into $\mathbb{R}^p$ with pairwise distances distorted at most ε, where $p = O(\varepsilon^{-2}\log|\mathscr{T}|)$ as one would expect from a JL embedding. This will be needed to guarantee the accuracy associated with our embeddings algorithms. Note that the bound on p does not require extra logarithmic factors, in contrast to the bound $O(\varepsilon^{-2}\log|\mathscr{T}|\log^4 n)$ in [21].

2.3 Sigma-Delta quantization

An r-th order $\Sigma\Delta$ quantizer $Q^{(r)} : \mathbb{R}^m \to \mathscr{A}^m$ maps an input signal $y = (y_i)_{i=1}^m \in \mathbb{R}^m$ to a quantized sequence $q = (q_i)_{i=1}^m \in \mathscr{A}^m$ via a quantization rule ρ and the following iterations

$$\begin{cases} u_0 = u_{-1} = \ldots = u_{1-r} = 0, \\ q_i = Q(\rho(y_i, u_{i-1}, \ldots, u_{i-r})) \quad \text{for } i = 1, 2, \ldots, m, \\ P^r u = y - q \end{cases} \tag{2.10}$$

where $Q(y) = \operatorname{argmin}_{v\in\mathscr{A}} |y-v|$ is the scalar quantizer related to alphabet $\mathscr{A}$ and $P \in \mathbb{R}^{m\times m}$ is the first order difference matrix defined by

$$P_{ij} = \begin{cases} 1 & \text{if } i = j, \\ -1 & \text{if } i = j+1, \\ 0 & \text{otherwise.} \end{cases}$$

Note that (2.10) is amenable to an iterative update of the state variables u_i as

$$P^r u = y - q \iff u_i = \sum_{j=1}^{r} (-1)^{j-1} \binom{r}{j} u_{i-j} + y_i - q_i, \quad i = 1,2,\ldots,m. \tag{2.11}$$

Definition 2.3.1. *A quantization scheme is* stable *if there exists $\mu > 0$ such that for each input with $\|y\|_\infty \le \mu$, the state vector $u \in \mathbb{R}^m$ satisfies $\|u\|_\infty \le C$. Crucially, μ and C do not depend on m.*

Stability heavily depends on the choice of quantization rule and is difficult to guarantee for arbitrary ρ in (2.10) when the alphabet is small, as is the case of 1-bit quantization where $\mathscr{A} = \{\pm 1\}$. When $r = 1$ and $\mathscr{A} = \{\pm 1\}$, the simplest stable $\Sigma\Delta$ scheme $Q^{(1)} : \mathbb{R}^m \to \mathscr{A}^m$ is equipped with the greedy quantization rule $\rho(y_i, u_{i-1}) := u_{i-1} + y_i$ giving the simple iteration (2.4) from the introduction, albeit with y_i replacing w_i.

A description of the design and properties of stable $Q^{(r)}$ with $r \ge 2$ can be found in Appendix 2.9.

2.4 Main Results

The ingredients that make our construction work are a JL embedding followed by $\Sigma\Delta$ quantization. Together these embed points into $\{\pm 1\}^m$, but it remains to define a pseudometric so that we may approximate Euclidean distances by distances on the cube. We now define this pseudometric.

Definition 2.4.1. *Let $\mathscr{A}^m = \{\pm 1\}^m$ and let $V \in \mathbb{R}^{p\times m}$ with $p \le m$. We define d_V on $\mathscr{A}^m \times \mathscr{A}^m$ as*

$$d_V(q_1,q_2) = \|V(q_1-q_2)\|_1 \quad \forall q_1,q_2 \in \mathscr{A}^m.$$

We now present our main result, a more technical version of Theorem 2.1.1, proved in Appendix 2.10.

Theorem 2.4.2 (Main result)**.** *Let λ, $r \in \mathbb{N}$, $\varepsilon \in (0,\frac{1}{2})$, $\delta \in (0,1)$, $\beta = \Omega(\log(|\mathscr{T}|/\delta)) > 0$, $\mu \in (0,1)$, $p = \Omega(\varepsilon^{-2}\log(|\mathscr{T}|^2/\delta)) \in \mathbb{N}$, and $m = \lambda p$. Let $\widetilde{V} \in \mathbb{R}^{p\times m}$ be as in Definition 2.2.3, $A \in \mathbb{R}^{m\times n}$ be the sparse Gaussian matrix in Definition 2.2.2 with $s = \Theta(\varepsilon^{-1}n^{-1}(\|v\|_\infty/\|v\|_2)^2) \le 1$, and Φ be the FJLT in Definition 2.2.4 with $s = \Theta(\varepsilon^{-1}n^{-1}(\|v\|_\infty/\|v\|_2)^2\log n) \le 1$.*

Let $\mathscr{T}$ be a finite subset of $B_2^n(\kappa) := \{x \in \mathbb{R}^n : \|x\|_2 \le \kappa\}$ and suppose that

$$\kappa \le \frac{\mu}{2\sqrt{\beta+\log(2m)}}.$$

Defining the embedding maps $f_1 : \mathscr{T} \to \{\pm 1\}^m$ by $f_1 = Q^{(r)} \circ A$ and $f_2 : \mathscr{T} \to \{\pm 1\}^m$ by $f_2 = Q^{(r)} \circ \Phi$, there exists a constant $C(\mu,r)$ such that the following are true:

(i) If the elements of $\mathscr{T}$ satisfy $\|x\|_\infty = O(n^{-1/2}\|x\|_2)$, then the bound

$$\left| d_{\widetilde{V}}(f_1(x),f_1(y)) - \|x-y\|_2 \right| \le C(\mu,r)\lambda^{-r+1/2} + \varepsilon\|x-y\|_2 \tag{2.12}$$

holds uniformly for all $x,y \in \mathscr{T}$ with probability exceeding $1-\delta-|\mathscr{T}|e^{-\beta}$.

(ii) On the other hand, for arbitrary $\mathscr{T} \subset B_2^n(\kappa)$

$$\left| d_{\widetilde{V}}(f_2(x),f_2(y)) - \|x-y\|_2 \right| \le C(\mu,r)\lambda^{-r+1/2} + \varepsilon\|x-y\|_2 \tag{2.13}$$

holds uniformly for any $x,y \in \mathscr{T}$ with probability exceeding $1-\delta-2|\mathscr{T}|e^{-\beta}$.

Under the assumptions of Theorem 2.4.2, we have

$$\varepsilon = O\Big(\sqrt{\frac{\log(|\mathscr{T}|^2/\delta)}{p}}\Big) \lesssim \frac{1}{\sqrt{p}}. \tag{2.14}$$

By (2.12), (2.13) and (2.14), we have that with high probability the inequality

$$\begin{aligned}\Big|d_{\widetilde{V}}(f_i(x),f_i(y)) - \|x-y\|_2\Big| &\le C(\mu,r)\Big(\frac{m}{p}\Big)^{-r+1/2} + \varepsilon\|x-y\|_2 \\ &\le C(\mu,r)\Big(\frac{m}{p}\Big)^{-r+1/2} + 2\kappa\varepsilon \\ &\le C(\mu,r)\Big(\frac{m}{p}\Big)^{-r+1/2} + \frac{\mu}{\sqrt{\beta+\log(2m)}}\cdot\frac{C_2}{\sqrt{p}}\end{aligned} \tag{2.15}$$

holds uniformly for $x,y \in \mathscr{T}$. The first error term in (2.15) results from $\Sigma\Delta$ quantization while the second error term is caused by the CJLT. So the term $O((m/p)^{-r+1/2})$ dominates when $\lambda = m/p$ is small. If m/p is sufficiently large, the second term $O(1/\sqrt{p})$ becomes dominant.

2.5 Computational and Space Complexity

In this section, we assume that $\mathscr{T} = \{x^{(j)}\}_{j=1}^k \subseteq \mathbb{R}^n$ consists of well-spread vectors. Moreover, we will focus on stable r-th order $\Sigma\Delta$ schemes $Q^{(r)} : \mathbb{R}^m \to \mathscr{A}^m$ with $\mathscr{A} = \{-1,1\}$. By Definition 2.2.3, when $r = 1$ we have $v = (1,1,\ldots,1) \in \mathbb{R}^\lambda$, while when $r = 2$, $v = (1,2,\ldots,\widetilde{\lambda}-1,\widetilde{\lambda},\widetilde{\lambda}-1,\ldots,2,1) \in \mathbb{R}^\lambda$. In general, $\|v\|_\infty/\|v\|_2 = O(\lambda^{-1/2})$ holds for all $r \in \mathbb{N}$. We also assume that $s = \Theta(\varepsilon^{-1}n^{-1}(\|v\|_\infty/\|v\|_2)^2) = \Theta(\varepsilon^{-1}n^{-1}\lambda^{-1}) \le 1$ as in Theorem 2.4.2. We consider b-bit floating-point or fixed-point representations for numbers. Both entail the same computational complexity for computing sums and products of two numbers. Addition and subtraction require $O(b)$ operations while multiplication and division require $\mathscr{M}(b) = O(b^2)$ operations via "standard" long multiplication and division. Multiplication and division can be done more efficiently, particularly for large integers and the best known methods (and best possible up to constants) have complexity $\mathscr{M}(b) = O(b\log b)$ [20]. We also assume random

access to the coordinates of our data points.

Embedding complexity. For each data point $x^{(j)} \in \mathscr{T}$, one can use Algorithm 1 to quantize it. Since A has sparsity constant $s = \Theta(\varepsilon^{-1}n^{-1}\lambda^{-1})$ and $\varepsilon^{-1} = O(p^{1/2})$ by (2.14), and since $\lambda = m/p$, computing $Ax^{(j)}$ needs $O(snm) = O(\lambda^{-1}\varepsilon^{-1}m) = O(p^{3/2})$ time. Additionally, it takes $O(m)$ time to quantize $Ax^{(j)}$ based on (2.21). When $p^{3/2} \leq m$, Algorithm 1 can be executed in $O(m)$ for each $x^{(j)}$. Because A has $O(snm) = O(m)$ nonzero entries, the space complexity is $O(m)$ bits per data point. Note that the big O notation here hides the space complexity dependence on the bit-depth b of the fixed or floating point representation of the entries of A and $x^{(j)}$. This clearly has no effect on the storage space needed for each $q^{(j)}$, which is exactly m bits.

Complexity of distance estimation. If one does not use embedding methods, storing $\mathscr{T}$ directly, i.e., by representing the coefficients of each $x^{(j)}$ by b bits requires knb bits. Moreover, the resulting computational complexity of estimating $\|x-y\|_2^2$ where $x, y \in \mathscr{T}$ is $O(n\mathscr{M}(b))$. On the other hand, suppose we obtain binary sequences $\mathscr{B} = \{q^{(j)}\}_{j=1}^{k} \subseteq \mathscr{A}^m$ by performing Algorithm 1 on $\mathscr{T}$. Using our method with accuracy guaranteed by Theorem 2.4.2, high-dimensional data points $\mathscr{T} \subseteq \mathbb{R}^n$ are now transformed into short binary sequences, which only require km bits of storage instead of knb bits. Algorithm 2 can be applied to recover the pairwise ℓ_2 distances. Note that $\widetilde{V}$ is the normalization of an integer valued matrix $V = I_p \otimes v$ (by Definition 2.2.3) and $q^{(i)} \in \mathscr{A}^m$ is a binary vector. So, by storing the normalization factor separately, we can ignore it when considering runtime and space complexity. Thus we observe:

1. The number of bits needed to represent each entry of v is at most $\log_2(\|v\|_\infty) \approx (r-1)\log_2\lambda = O(\log_2\lambda)$ when $r > 1$ and $O(1)$ when $r = 1$. So the computation of $y^{(i)} = \widetilde{V}q^{(i)} \in \mathbb{R}^p$ only involves m additions or subtractions of integers represented by $O(\log_2\lambda)$ bits and thus the time complexity in computing $y^{(i)}$ is $O(m\log_2\lambda)$.

2. Each of the p entries of $y^{(i)}$ is the sum of λ terms each bounded by λ^{r-1}. We can store $y^{(i)}$ in $O(p\log_2\lambda)$ bits.

3. Computing $\|y^{(i)} - y^{(j)}\|_1$ needs $O(p\log_2\lambda)$ time and $O(p\log_2\lambda)$ bits.

So we use $O(p\log_2\lambda)$ bits to recover each pairwise distance $\|x^{(i)}-x^{(j)}\|_2$ in $O(m\log_2\lambda)$ time.

Table 2.1. Here "Time" is the time needed to embed a data point, while "Space" is the space needed to store the embedding matrix. "Storage" contains the memory usage to store each encoded sequence. "Query time" is the time complexity of pairwise distance estimation.

Method	Time	Space	Storage	Query Time
Gaussian Toeplitz [40]	$O(n\log n)$	$O(n)$	$O(m)$	$O(m)$
Bilinear [15]	$O(n\sqrt{m})$	$O(\sqrt{mn})$	$O(m)$	$O(m)$
Circulant [41]	$O(n\log n)$	$O(n)$	$O(m)$	$O(m)$
BOE or PCE* [21]	$O(n\log n)$	$O(n)$	$O(p\log_2\lambda)$	$O(p\mathscr{M}(\log_2\lambda))$
Our Algorithm*	$O(m)$	$O(m)$	$O(p\log_2\lambda)$	$O(p\log_2\lambda)$

* These algorithms recover Euclidean distances and others recover geodesic distances.

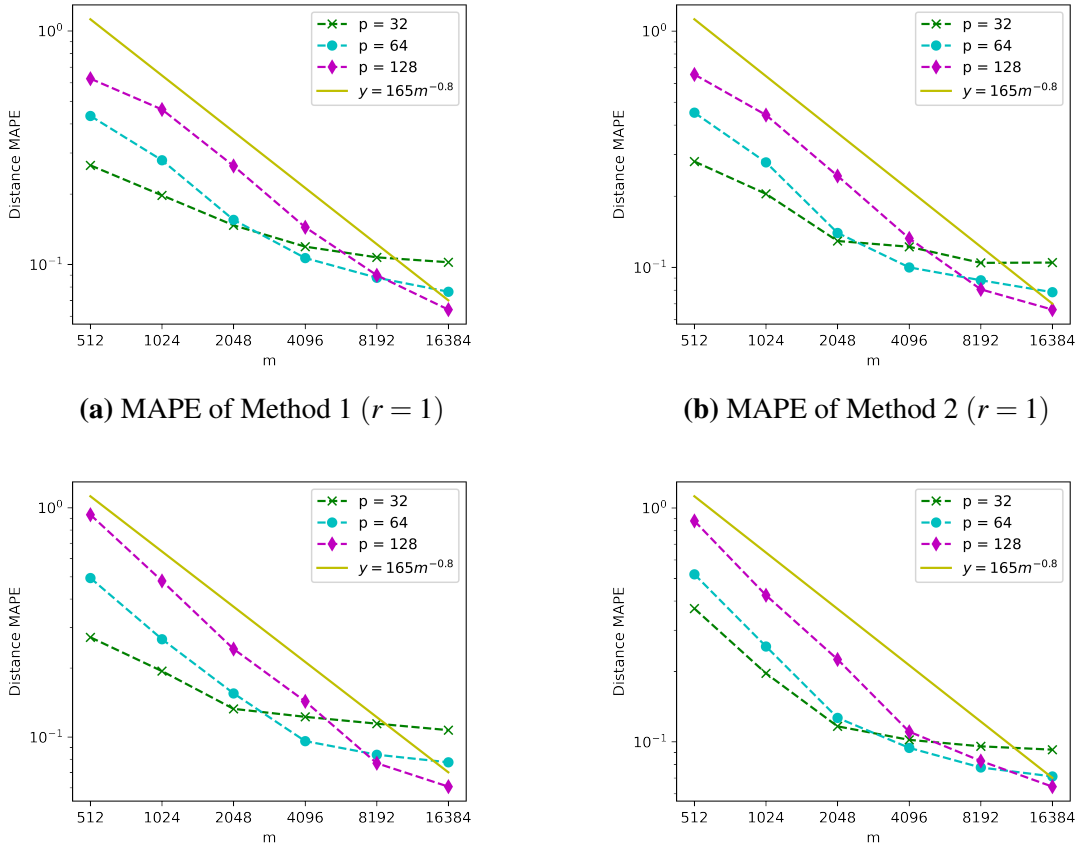
Comparisons with baselines. In Table 2.1, we compare our algorithm with various JL-based methods from Section 2.1. Here n is the input dimension, m is the embedding dimension (and number of bits), and $p=m/\lambda$ is the length of encoded sequences $y=\widetilde{V}q$. In our case, we use $O(p\log_2\lambda)$ to store $y=\widetilde{V}q$. See Appendix 2.11 for a comparison with product quantization.

2.6 Numerical Experiments

To illustrate the performance of our fast binary embedding (Algorithm 1) and ℓ_2 distance recovery (Algorithm 2), we apply them to real-world datasets: Yelp open dataset[1], ImageNet [10], Flickr30k [34], and CIFAR-10 [28]. All images are converted to grayscale and resampled using bicubic interpolation to size 128×128 for images from Yelp, ImageNet, and Flickr30k and 32×32 for images from CIFAR-10. So, each can be represented by a 16384-dimensional or 1024-dimensional vector. The results are reported here and in Appendix 2.7. We consider the two versions of our fast binary embedding algorithm from Theorem 2.4.2:

Method 1. Quantize FJLT embeddings Φx and recover distances based on Algorithm 2.

[1] Yelp open dataset: https://www.yelp.com/dataset



(a) MAPE of Method 1 $(r = 1)$

(b) MAPE of Method 2 $(r = 1)$

(c) MAPE of Method 1 $(r = 2)$

(d) MAPE of Method 2 $(r = 2)$

Figure 2.1. Plots of ℓ_2 distance reconstruction error when $r = 1, 2$

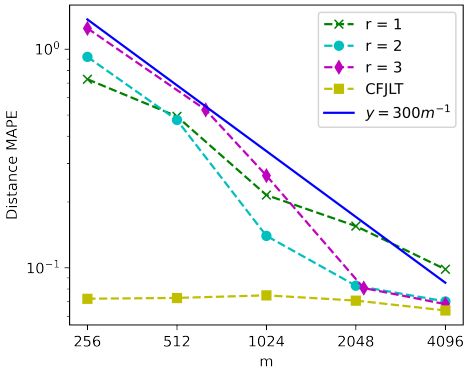
Method 2. Quantize sparse JL embeddings Ax and recover distances by Algorithm 2.

In order to test the performance of our algorithm, we compute the mean absolute percentage error (MAPE) of reconstructed ℓ_2 distances averaged over all pairwise data points, that is,

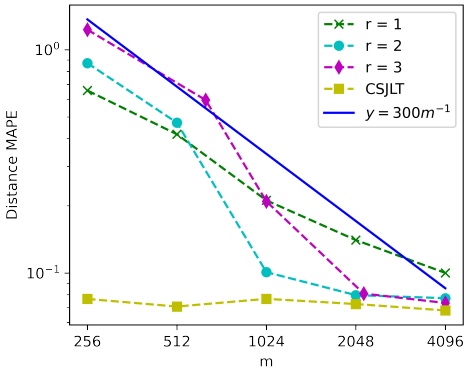
$$\frac{2}{k(k-1)} \sum_{x,y \in \mathscr{T}} \left| \frac{\|\widetilde{V}(q_x - q_y)\|_1 - \|x-y\|_2}{\|x-y\|_2} \right|.$$

Experiments on the Yelp dataset. To give a numerical illustration of the relation among the length m of the binary sequences, embedding dimension p, and order r, as compared to the upper bound in (2.15), we use both Method 1 and Method 2 on the Yelp dataset. We randomly sample $k = 1000$ images and scale them by the same constant so all data points are contained in the

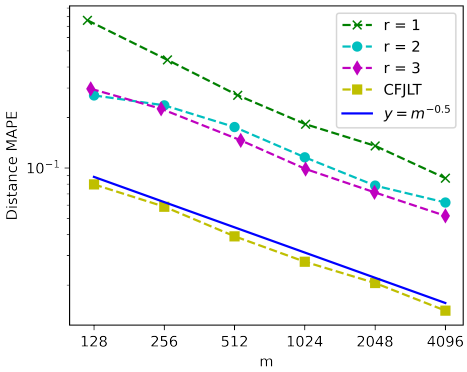
ℓ_2 unit ball. The scaled dataset is denoted by $\mathscr{T}$. Based on Theorem 2.4.2, we set $n = 16384$ and $s = 1650/n \approx 0.1$. For each fixed p, we apply Algorithm 1 and Algorithm 2 for various m. We present our experimental results for stable $\Sigma\Delta$ quantization schemes, given by (2.21), with $r = 1$ and $r = 2$ in Figure 2.1. For $r = 1$, we observe that the curve with small p quickly reaches an error floor while with high p the error decays like $m^{-1/2}$ and eventually reach a lower floor. The reason is that the first error term in (2.15) is dominant when m/p is relatively small but the second error term eventually dominates as m becomes larger and larger. When $r = 2$ the error curves decay faster and eventually achieve the same flat error because now the first term in (2.15) has power $-3/2$ while the second flat error term is independent of r. Moreover, the performance of Method 2 is very similar to that of Method 1.



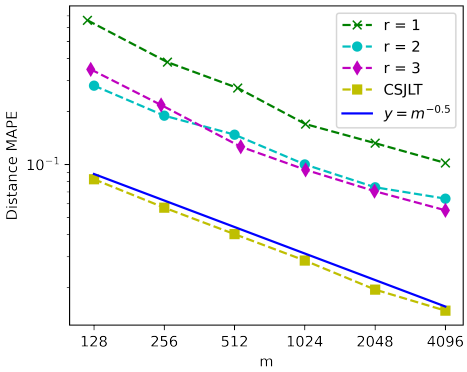
(a) MAPE of Method 1 ($p = 64$)



(b) MAPE of Method 2 ($p = 64$)



(c) MAPE of Method 1 ($p = p(m)$)



(d) MAPE of Method 2 ($p = p(m)$)

Figure 2.2. Plots of ℓ_2 distance reconstruction error with fixed $p = 64$ and optimal $p = p(m)$

Next, we illustrate the relationship between the quantization order r and the number of measurements m in Figure 2.2. The curves obtained directly from an unquantized CFJLT (resp. CSJLT) as in Lemma 2.2.6, with $m = 256, 512, 1024, 2048, 4096$, and $p = 64$ are used for comparison against the quantization methods. The first row of Figure 2.2 depicts the mean squared relative error when $p = 64$ is fixed for all distinct methods. It shows that stable quantization schemes with order $r > 1$ outperform the first order greedy quantization method, particularly when m is large. Moreover, both the $r = 2$ and $r = 3$ curves converge to the CFJLT/CSJLT result as m goes to 4096. Note that by using a quarter of the original dimension, i.e. $m = 4096$, our construction achieves less than 10% error. Furthermore, if we encode $\widetilde{V}q$ as discussed in Section 2.5, then we need at most $rp\log_2\lambda = 64r\log_2(4096/64) = 384r$ bits per image, which is $\lesssim 0.023$ bits per pixel.

For our final experiment, we illustrate that the performance of the proposed approach can be further improved. Note that the choice of p only affects the distance computation in Algorithm 2 and does not appear in the embedding algorithm. In other words, one can vary p in Algorithm 2 to improve performance. This can be done either analytically by viewing the right hand side of (2.15) as a function of p and optimizing for p (up to constants). It can also be done empirically, as we do here. Following this intuition, if we vary p as a function of m, and use the empirically optimal $p := p(m)$ in the construction of $\widetilde{V}$, then we obtain the second row of Figure 2.2 where the choice $r = 3$ exhibits lower error than other quantization methods. Note that the decay rate, as a function of m, very closely resembles that of the unquantized JL embedding particularly for higher orders r (as one can verify by optimizing the right hand side of (2.15)).

2.7 Comparisons on different datasets

Experiments on the Yelp dataset in Section 2.6 showed that Method 2 based on sparse JL embeddings performs as well as Method 1 which usues an FJLT to enforce the well-spreadness assumption. Now, we only focus on Method 2 and check its performance on all four different

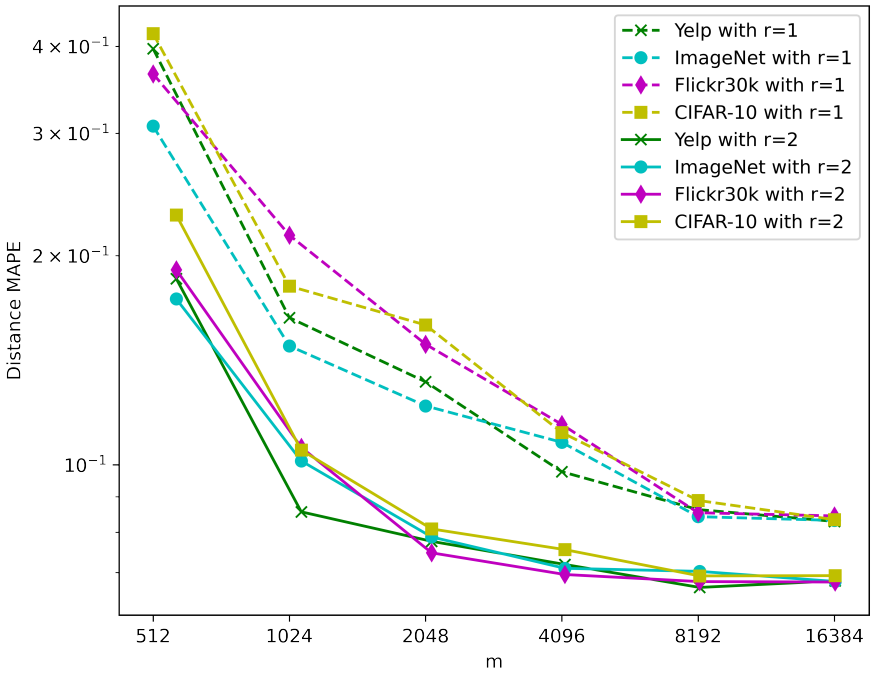


Figure 2.3. Plot of MAPE of Method 2 on four datasets with fixed $p = 64$ and order $r = 1, 2$

datasets: Yelp, ImageNet, Flickr30k, and CIFAR-10.

Specifically, for each dataset we randomly sample $k = 1000$ images and scale them such that all scaled data points are contained in the ℓ_2 unit ball. Then we apply Method 2 to each dataset separately and compute the corresponding MAPE metric, see Figure 2.3, where we fix $p = 64$ and let $r = 1, 2$. We can observe that curves with $r = 1$ fluctuate, but displays a clear downward trend, when $m \leq 8192$ and reach an error floor around 0.08. In contrast to the first order quantization scheme, curves with $r = 2$ decays faster and eventually achieve a lower floor around 0.07. Additionally, Method 2 performs well on all datasets and implies that assumption of well-spread input vectors is not too restrictive on natural images.

2.8 Proof of Lemma 2.2.6

We will require the following lemmas, adapted from the literature, to prove the distance-preserving properties of our condensed sparse Johnson-Lindenstrauss transform (CSJLT) and condensed fast Johnson-Lindenstrauss transform (CFJLT) in Lemma 2.2.6.

Lemma 2.8.1 (Theorem 5.1 in [31])**.** *Let $n \in \mathbb{N}$, $\varepsilon \in (0, \frac{1}{2})$, $\delta \in (0,1)$, $\alpha \in [\frac{1}{\sqrt{n}}, 1]$ be parameters and set $m = C\varepsilon^{-2}\log(\delta^{-1}) \in \mathbb{N}$ where C is a sufficiently large constant. Let $s = 2\alpha^2/\varepsilon \leq 1$, $A \in \mathbb{R}^{m\times n}$ be as in Definition 2.2.2. Then*

$$P\Big((1-\varepsilon)\|x\|_2 \leq \frac{\sqrt{\pi/2}}{m}\|Ax\|_1 \leq (1+\varepsilon)\|x\|_2\Big) \geq 1-\delta \tag{2.16}$$

holds for all $x \in \mathbb{R}^n$ with $\|x\|_\infty \leq \alpha\|x\|_2$.

Lemma 2.8.2 below is adapted from [1, Lemma 1], and we present its proof for completeness.

Lemma 2.8.2. *Let $H \in \mathbb{R}^{n\times n}$ and $D \in \mathbb{R}^{n\times n}$ be as in Definition 2.2.4. For any $\lambda > 0$ and $x \in \mathbb{R}^n$ we have*

$$P\Big(\|HDx\|_\infty \leq \lambda\|x\|_2\Big) \geq 1 - 2ne^{-n\lambda^2/2}. \tag{2.17}$$

Proof. Without loss of generality, we can assume $\|x\|_2 = 1$. Let $u = HDx = (u_1, \ldots, u_n)$. Fix $i \in \{1, \ldots, n\}$. Then $u_i = \sum_{j=1}^n a_j x_j$ with $P\Big(a_j = \frac{1}{\sqrt{n}}\Big) = P\Big(a_j = -\frac{1}{\sqrt{n}}\Big) = \frac{1}{2}$ for all j. Moreover, $a_1, a_2, \ldots, a_n$ are independent and symmetric. So u_i is also symmetric, that is, u_i and $-u_i$ share the same distribution. For any $t \in \mathbb{R}$ we have

$$\begin{aligned}\mathbb{E}(e^{tnu_i}) = \prod_{j=1}^n \mathbb{E}[\exp(tna_jx_j)] &= \prod_{j=1}^n \frac{\exp(t\sqrt{n}x_j) + \exp(-t\sqrt{n}x_j)}{2} \\ &\leq \prod_{j=1}^n \exp(nt^2x_j^2/2) = \exp(nt^2/2).\end{aligned}$$

Since u_i is symmetric, by Markov's inequality and the above result, we get

$$P(|u_i| \geq \lambda) = 2P(e^{\lambda n u_i} \geq e^{\lambda^2 n}) \leq 2e^{-\lambda^2 n}\mathbb{E}(e^{\lambda n u_i}) = 2e^{-\lambda^2 n/2}.$$

Inequality (2.17) follows by the union bound over all $i \in \{1,\ldots,n\}$. □

Lemma 2.8.3. *Let $n, \lambda \in \mathbb{N}$, $\varepsilon \in (0, \frac{1}{2})$, $\delta \in (0,1)$, $p = O(\varepsilon^{-2}\log(\delta^{-1})) \in \mathbb{N}$ and $m = \lambda p$. Let $\widetilde{V} \in \mathbb{R}^{p\times m}$ be as in Definition 2.2.3, $A \in \mathbb{R}^{m\times n}$ be the sparse Gaussian matrix in Definition 2.2.2 with $s = \Theta(\varepsilon^{-1}n^{-1}(\|v\|_\infty/\|v\|_2)^2) \leq 1$, and $\Phi = AHD \in \mathbb{R}^{m\times n}$ be the FJLT in Definition 2.2.4 with $s = \Theta(\varepsilon^{-1}n^{-1}(\|v\|_\infty/\|v\|_2)^2\log n) \leq 1$. Then for $x \in \mathbb{R}^n$ with $\|x\|_\infty = O(n^{-1/2}\|x\|_2)$, we have*

$$\mathrm{P}\Big((1-\varepsilon)\|x\|_2 \leq \|\widetilde{V}Ax\|_1 \leq (1+\varepsilon)\|x\|_2\Big) \geq 1-\delta, \tag{2.18}$$

and for arbitrary $x \in \mathbb{R}^n$, we have

$$\mathrm{P}\Big((1-\varepsilon)\|x\|_2 \leq \|\widetilde{V}\Phi x\|_1 \leq (1+\varepsilon)\|x\|_2\Big) \geq 1-\delta. \tag{2.19}$$

Proof. Recall that $V = I_p \otimes v$ and $\Phi = AHD$. Let $y \in \mathbb{R}^n$ and $K := VA = (I_p \otimes v)A \in \mathbb{R}^{p\times n}$. For $1 \leq i \leq p$ and $1 \leq j \leq n$, we have

$$K_{ij} = \sum_{k=1}^{\lambda} v_k a_{(i-1)\lambda+k,j}.$$

Denote the row vectors of A by $a_1, a_2, \ldots, a_m$. It follows that

$$(Ky)_i = \sum_{j=1}^{n} K_{ij}y_j = \sum_{j=1}^{n}\sum_{k=1}^{\lambda} y_j v_k a_{(i-1)\lambda+k,j} = \sum_{k=1}^{\lambda} v_k\langle y, a_{(i-1)\lambda+k}\rangle = [B(v^{\mathsf{T}} \otimes y)]_i$$

where

$$B := \begin{bmatrix} a_1 & a_2 & \dots & a_\lambda \\ a_{\lambda+1} & a_{\lambda+2} & \dots & a_{2\lambda} \\ \vdots & \vdots & & \vdots \\ a_{(p-1)\lambda+1} & a_{(p-1)\lambda+2} & \dots & a_{p\lambda} \end{bmatrix} \in \mathbb{R}^{p\times\lambda n} \quad \text{and} \quad v^{\mathsf{T}} \otimes y = \begin{bmatrix} v_1 y \\ v_2 y \\ \vdots \\ v_\lambda y \end{bmatrix} \in \mathbb{R}^{\lambda n}.$$

Hence $VAy = Ky = B(v^{\mathsf{T}} \otimes y)$ holds for all $y \in \mathbb{R}^n$. Additionally, we get a reshaped sparse Gaussian random matrix B by rearranging the rows of A.

For the first assertion in the theorem, note that $x \in \mathbb{R}^n$ satisfies $\|x\|_\infty = O(\|x\|_2/\sqrt{n})$. So, we have $VAx = B(v^{\mathsf{T}} \otimes x)$, $\|v^{\mathsf{T}} \otimes x\|_2 = \|v\|_2\|x\|_2$ and $\|v^{\mathsf{T}} \otimes x\|_\infty = \|v\|_\infty\|x\|_\infty$. Then (2.18) holds by applying Lemma 2.8.1 to random matrix B and vector $v^\top \otimes x$ with $\alpha = \Theta(n^{-1/2}\|v\|_\infty/\|v\|_2)$.

For the second assertion, if $x \in \mathbb{R}^n$ is arbitrary, then by substituting HDx for y one can get $V\Phi x = B(v^{\mathsf{T}} \otimes (HDx))$. Note that $\|v^{\mathsf{T}} \otimes (HDx)\|_2 = \|v\|_2\|HDx\|_2 = \|v\|_2\|x\|_2$ and $\|v^{\mathsf{T}} \otimes (HDx)\|_\infty = \|v\|_\infty\|HDx\|_\infty$. Inequality (2.19) follows immediately by using the above fact and applying Lemma 2.8.1 and Lemma 2.8.2 to the random operator B and vector $v^{\mathsf{T}} \otimes (HDx)$ with $\alpha = \Theta((n^{-1}\log n)^{1/2}\|v\|_\infty/\|v\|_2)$. □

Now we can embed a set of points in a high dimensional space into a space of much lower dimension in such a way that distances between the points are nearly preserved. By substituting δ with $2\delta/|\mathscr{T}|^2$ in Lemma 2.8.3 and using the fact $1 - \binom{|\mathscr{T}|}{2}\frac{2\delta}{|\mathscr{T}|^2} = 1 - \frac{|\mathscr{T}|(|\mathscr{T}|-1)}{2} \cdot \frac{2\delta}{|\mathscr{T}|^2} > 1-\delta$, Lemma 2.2.6 follows from the union bound over all pairwise data points in $\mathscr{T}$.

2.9 Stable Sigma-Delta quantization and its properties

Although it is a non-trivial task to design a stable quantization rule ρ when $r > 1$, families of one-bit $\Sigma\Delta$ quantization schemes that achieve this goal have been designed by [8, 19, 9], and we now describe one such family. To start, note that an r-th order $\Sigma\Delta$ quantization scheme may

also arise from a more general difference equation of the form

$$y - q = f * v \tag{2.20}$$

where $*$ denotes convolution and the sequence $f = P^r g$ with $g \in \ell^1$. Then any (bounded) solution v of (2.20) generates a (bounded) solution u of (2.11) via $u = g * v$. Thus (2.11) can be rewritten in the form (2.20) by a change of variables. Defining $h := \delta^{(0)} - f$, where $\delta^{(0)}$ denotes the Kronecker delta sequence supported at 0, and choosing the quantization rule ρ in terms of the new variable as $(h*v)_i + y_i$. Then (2.10) reads as

$$\begin{cases} q_i = Q((h*v)_i + y_i), \\ v_i = (h*v)_i + y_i - q_i. \end{cases} \tag{2.21}$$

By designing a proper filter h one can get a stable r-th order $\Sigma\Delta$ quantizer, as was done in [9, 19], leading to the following result from [19], which exploits the above relationship between v and u to bound $\|u\|_\infty$.

Proposition 2.9.1. *Fix an integer r, an integer $\sigma \geq 6$ and let $n_j = \sigma(j-1)^2 + 1$ for $j = 1, 2, \ldots, r$. Let the filter h be of the form*

$$h = \sum_{j=1}^{r} d_j \delta^{n_j}$$

where δ^{n_j} is the Kronecker delta supported at n_j and $d_j = \prod_{i \neq j} \frac{n_i}{n_i - n_j}$ for $j = 1, 2, \ldots, r$. There exists a universal constant $C > 0$ such that the rth order $\Sigma\Delta$ scheme (2.21) *with* 1*-bit alphabet $\mathscr{A} = \{-1, 1\}$, is stable, and*

$$\|y\|_\infty \leq \mu < 1 \Longrightarrow \|u\|_\infty \leq Cc(\mu)^r r^r, \tag{2.22}$$

where $c(\mu) > 0$ is a constant only depends on μ.

Having introduced stable $\Sigma\Delta$ quantization, we now present a lemma controlling an

operator norm of $\widetilde{V}P^r$. We will need this result in controlling the error in approximating distances associated with our binary embedding.

Lemma 2.9.2. *For a stable r-th order ΣΔ quantization scheme,*

$$\|\widetilde{V}P^r\|_{\infty,1} \leq \sqrt{\pi/2}(8r)^{r+1}\lambda^{-r+1/2}.$$

Proof. By the same method used in the proof of Lemma 4.6 in [21], one can get

$$\|VP^r\|_{\infty,\infty} \leq r2^{3r-1} \quad \text{and} \quad \|v\|_2 \geq \lambda^{r-1/2}r^{-r}.$$

It follows that

$$\|\widetilde{V}P^r\|_{\infty,1} = \frac{\sqrt{\pi/2}}{p\|v\|_2}\|VP^r\|_{\infty,1} \leq \frac{\sqrt{\pi/2}}{\|v\|_2}\|VP^r\|_{\infty,\infty} \leq \sqrt{\pi/2}(8r)^{r+1}\lambda^{-r+1/2}.$$

□

The following result guarantees that the linear part of our embedding generates a bounded vector, and therefore allows us to later appeal to the stability property of ΣΔ quantizers. In other words, it will allow us to use (2.22) to control the infinity norm of state vectors generated by ΣΔ quantization.

Lemma 2.9.3 (Concentration inequality for $\|\cdot\|_\infty$)**.** *Let $\beta > 0$, $\varepsilon \in (0,1)$, $A \in \mathbb{R}^{m\times n}$ be the sparse Gaussian matrix in Definition 2.2.2 with $s = \Theta(\varepsilon^{-1}n^{-1}) \leq 1$, and $\Phi = AHD \in \mathbb{R}^{m\times n}$ be the FJLT in Definition 2.2.4 with $s = \Theta(\varepsilon^{-1}n^{-1}\log n) \leq 1$. Suppose that*

$$2\sqrt{\beta + \log(2m)} \leq \mu \leq \frac{4}{\sqrt{\varepsilon}}. \tag{2.23}$$

Then

$$P\big(\|Ax\|_\infty \leq \mu\|x\|_2\big) \geq 1 - e^{-\beta} \tag{2.24}$$

holds for $x \in \mathbb{R}^n$ *with* $\|x\|_\infty = O(n^{-1/2}\|x\|_2)$ *and*

$$P\big(\|\Phi x\|_\infty \le \mu \|x\|_2\big) \ge 1 - 2e^{-\beta} \tag{2.25}$$

holds for $x \in \mathbb{R}^n$.

Proof. Without loss of generality, we can assume that x is a unit vector with $\|x\|_2 = 1$. We start with the proof of (2.25). By applying Lemma 2.8.2 to x with $\lambda = \Theta(\sqrt{\log n/n})$, we have

$$P\Big(\|HDx\|_\infty \le \lambda\Big) \ge 1 - e^{-\beta}. \tag{2.26}$$

Let A be as in Definition 2.2.2 with $s = 2\lambda^2/\varepsilon = \Theta(\varepsilon^{-1}n^{-1}\log n) \le 1$ and recall that $\Phi = AHD$.

Suppose that $y \in \mathbb{R}^n$ with $\|y\|_2 = 1$ and $\|y\|_\infty \le \lambda$. Let $Y = Ay$. Then $Y_i := (Ay)_i = \sum_{j=1}^n a_{ij}y_j$ for $1 \le i \le m$. For $t \le t_0 := \sqrt{2s}/\lambda = 2/\sqrt{\varepsilon}$, we get $t^2y_j^2/2s \le 1$ for all j. Since $e^x \le 1+2x$ for all $x \in [0,1]$ and $1+x \le e^x$ for all $x \in \mathbb{R}$, $se^{t^2y_j^2/2s} + 1 - s \le s(1+t^2y_j^2/s) + 1 - s = 1 + t^2y_j^2 \le e^{t^2y_j^2}$.

It follows that

$$\mathbb{E}(e^{tY_i}) = \prod_{j=1}^n \mathbb{E}(e^{ta_{ij}y_j}) = \prod_{j=1}^n \big(se^{t^2y_j^2/2s} + 1 - s\big) \le \prod_{j=1}^n e^{t^2y_j^2} = e^{t^2}$$

holds for all $1 \le i \le m$ and $t \in [0, t_0]$. So for $t \in [0, t_0]$, by Markov inequality and above inequality we have

$$P\Big(Y_i \ge \mu\Big) = P\Big(e^{tY_i} \ge e^{t\mu}\Big) \le e^{-t\mu}\mathbb{E}(e^{tY_i}) \le e^{-t\mu + t^2}.$$

According to (2.23) we can set $t = \mu/2 \le t_0 = 2/\sqrt{\varepsilon}$, then $P\Big(Y_i \ge \mu\Big) \le e^{-\mu^2/4}$. By symmetry we have $P\Big(-Y_i \ge \mu\Big) \le e^{-\mu^2/4}$. Consequently, for all $1 \le i \le m$ we have

$$P\Big(|Y_i| \ge \mu\Big) \le 2e^{-\mu^2/4}. \tag{2.27}$$

By a union bound, (2.23), and (2.27)

$$P\big(\|Ay\|_\infty \geq \mu\big) = P\Big(\max_{1\leq i\leq m} |Y_i| \geq \mu\Big) \leq mP\Big(|Y_i| \geq \mu\Big)$$
$$= 2me^{-\mu^2/4} \leq e^{-\beta}. \tag{2.28}$$

It follows immediately from (2.26) and (2.28) with $y = HDx$ that

$$\begin{aligned} P(\|\Phi x\|_\infty \leq \mu) &= P(\|AHDx\|_\infty \leq \mu) \\ &\geq P(\|AHDx\|_\infty \leq \mu, \|HDx\|_\infty \leq \lambda) \\ &= P(\|AHDx\|_\infty \leq \mu \mid \|HDx\|_\infty \leq \lambda) P(\|HDx\|_\infty \leq \lambda) \\ &\geq (1-e^{-\beta})^2 \\ &\geq 1-2e^{-\beta}. \end{aligned}$$

Furthermore, if we replace y by x in (2.28) and use A with $s = \Theta(\varepsilon^{-1}n^{-1})$, then inequality (2.24) follows. The difference in the choice of s is due to the fact that for vectors in the unit ball with $\|x\|_\infty = O(n^{-1/2}\|x\|_2)$ we have that $\|x\|_\infty \leq n^{-1/2}$. □

2.10 Proof of Theorem 2.4.2

Proof. Since the proofs of (2.12) and (2.13) are almost identical except for using different random projections A and Φ, we shall only establish the result for (2.13) in detail. For any $x \in \mathscr{T} \subseteq B_2^n(\kappa)$ we have $\|x\|_2 \leq \kappa$. By applying Lemma 2.9.3 we get

$$\begin{aligned} P\big(\|\Phi x\|_\infty < \mu\big) &\geq P\big(\|\Phi x\|_\infty < \mu\|x\|_2/\kappa\big) \\ &\geq P\big(\|\Phi x\|_\infty < 2\sqrt{\beta+\log(2m)}\|x\|_2\big) \\ &\geq 1-2e^{-\beta}. \end{aligned}$$

Since above inequality holds for arbitrary $x \in \mathscr{T}$, by union bound one can get

$$P\Big(\max_{x\in\mathscr{T}} \|\Phi x\|_\infty < \mu\Big) \geq 1 - 2|\mathscr{T}|e^{-\beta}.$$

Suppose that u_x is the state vector of input signal Φx which is produced by stable r-th order $\Sigma\Delta$ scheme. Using Lemma 2.9.2 and formula (2.22) to get

$$\|\widetilde{V}P^r\|_{\infty,1}\|u_x\|_\infty \leq Cc(\mu)^r r^r (8r)^{r+1}\sqrt{\pi/2}\lambda^{-r+1/2}, \tag{2.29}$$

which holds uniformly for all $x \in \mathscr{T}$ with probability exceeding $1 - 2|\mathscr{T}|e^{-\beta}$.

Furthermore, by Lemma 2.2.6 the probability that

$$\Big|\|\widetilde{V}\Phi(x-y)\|_1 - \|x-y\|_2\Big| \leq \varepsilon\|x-y\|_2 \tag{2.30}$$

holds simultaneously for all $x, y \in \mathscr{T}$ is at least $1-\delta$.

We deduce from triangle inequality and equations (2.29), (2.30) that

$$\begin{aligned}
&\Big|d_{\widetilde{V}}(f_2(x), f_2(y)) - \|x-y\|_2\Big| \\
&= \Big|\|\widetilde{V}Q^{(r)}(\Phi x) - \widetilde{V}Q^{(r)}(\Phi y)\|_1 - \|x-y\|_2\Big| \\
&\leq \Big|\|\widetilde{V}Q^{(r)}(\Phi x) - \widetilde{V}Q^{(r)}(\Phi y)\|_1 - \|\widetilde{V}\Phi(x-y)\|_1\Big| + \Big|\|\widetilde{V}\Phi(x-y)\|_1 - \|x-y\|_2\Big| \\
&\leq \|\widetilde{V}(Q^{(r)}(\Phi x) - \Phi x) - \widetilde{V}(Q^{(r)}(\Phi y) - \Phi y)\|_1 + \Big|\|\widetilde{V}\Phi(x-y)\|_1 - \|x-y\|_2\Big| \\
&\leq \|\widetilde{V}P^r u_x\|_1 + \|\widetilde{V}P^r u_y\|_1 + \Big|\|\widetilde{V}\Phi(x-y)\|_1 - \|x-y\|_2\Big| \\
&\leq \|\widetilde{V}P^r\|_{\infty,1}(\|u_x\|_\infty + \|u_y\|_\infty) + \Big|\|\widetilde{V}\Phi(x-y)\|_1 - \|x-y\|_2\Big| \\
&\leq 2Cc(\mu)^r r^r (8r)^{r+1}\sqrt{\pi/2}\lambda^{-r+1/2} + \varepsilon\|x-y\|_2 \\
&= \sqrt{2\pi}Cc(\mu)^r r^r (8r)^{r+1}\lambda^{-r+1/2} + \varepsilon\|x-y\|_2 \\
&= C(\mu,r)\lambda^{-r+1/2} + \varepsilon\|x-y\|_2
\end{aligned}$$

holds uniformly for any $x, y \in \mathscr{T}$ with probability at least $1 - \delta - 2|\mathscr{T}|e^{-\beta}$. The bound (2.12) is associated with a weaker condition on β due to the associated weaker condition in Lemma 2.9.3. □

2.11 Comparison with product quantization

Note that the distance preserving quality (as well as performance on retrieval and classification tasks) of MSQ binary embeddings using bilinear projection [15] or circulant matrices [41] has be shown to be at least as good as product quantization [22], LSH [3, 38] and ITQ [18]. Our method uses Sigma-Delta quantization, which

1. gives provably better error rates than the MSQ design as shown in this paper, and in [21];
2. is more efficient in terms of both memory and distance query computation as shown in Section 2.5.

In order to more explicitly compare our algorithm with data dependent methods, as an example, we now briefly analyze product quantization as presented in [22]. We then present a brief analysis of optimal data-independent methods as well as data-independent product quantization, in comparison with our method.

2.11.1 Data-dependent product quantization

The key idea here is to decompose the input vector space $\mathbb{R}^n$ into the Cartesian product of M low-dimensional subspaces $\mathbb{R}^d$ with $n = Md$ and quantize each subspace into k^* codewords, for example by using the k-means algorithm. So the total number of centroids (codewords) in $\mathbb{R}^n$ is $k = (k^*)^M$ and the time complexity of learning all k centroids is $O(nNk^*t)$ where N is the number of training data points and t is the number of iterations in the k-means algorithm. Moreover, converting each input vector $x \in \mathbb{R}^n$ to the index of its codeword needs time $O(Mdk^*) = O(nk^*)$ and the length of binary codes is $m = \log_2 k = M \log_2 k^*$. Since we have to store all k centroids and M lookup tables, memory usage is $O(M(dk^* + (k^*)^2)) = O(nk^* + M(k^*)^2)$. Moreover, the

query time, i.e. the time complexity of pairwise distance estimation is $O(Mk^*)$ using lookup tables. As a result, we obtain Table 2.2, whose column headings are analogous to those in Table 2.1.

Table 2.2. Comparison between the proposed method and product quantization per data point

Method	Time	Space	Storage	Query Time
Product Quantization	$O(nk^*)$	$O(nk^* + M(k^*)^2)$	$O(M \log_2 k^*)$	$O(Mk^*)$
Our Method (on well-spread $\mathscr{T}$)	$O(m)$	$O(m)$	$O(p \log_2 \lambda)$	$O(p \log_2 \lambda)$

A direct comparison of the associated errors is not possible due to the fact that the error associated with data-dependent product quantization is a function of the input data distribution, and the convergence of the k-means algorithm. Nevertheless, one can note some tradeoffs from Table 2.2. Namely, the embedding time and the space needed to store our embedding matrix are lower than those associated with product quantization. On the other hand, the space needed to store the embedded data points and the query time associated with product quantization depend on the parameter choices M and k^*, which also affect the resulting accuracy. Finally, we note that product quantization (using k-means clustering) is associated with a pre-processing time $O(nNk^*t)$, which is significantly larger than our method.

Data-independent product quantization and optimality of our method

If one were to just encode, in a data independent way, the ℓ_2 ball of $\mathbb{R}^n$, so that the encoding error is at most θ, then a simple volume argument shows that one needs at least θ^{-n} codewords, hence $n \log_2(1/\theta)$ bits. This lower bound holds, independent of the encoding method, i.e., whether one uses product quantization or any other technique. To reduce the number of bits below n, one approach is to capitalize on the finiteness of the data, and use a JL type embedding (such as random sampling for well-spread data) to reduce the dimension to $p \approx \log|T|/\varepsilon^2$ (up to log factors), and therefore introduce a new embedding error of ε, on top of the encoding error.

The advantage is that one would then only need to encode an ℓ_2 ball in the p-dimensional space. Again, independently of the encoding method, one would now need $p\log(1/\theta)$ bits to get an encoding error of θ. If we denote c_x, c_y, the encoding of x and y, then this gives the error estimate

$$\big|\|c_x - c_y\| - \|x-y\|\big| \lesssim \theta + \varepsilon\|x-y\|.$$

If we rewrite the error now in terms of the number of bits $b = p\log(1/\theta)$, we get

$$\big|\|c_x - c_y\| - \|x-y\|\big| \lesssim 2^{-b/p} + \varepsilon\|x-y\|.$$

Note that in all of this, no computational complexity was taken into account.

One can envision replacing k-means clustering in product quantization, with a data-independent encoding. With a careful choice of parameters, this may be significantly more computationally efficient than the above optimal encoding, albeit at the expense of a sub-optimal error bound.

On the other hand, consider that our computationally efficient scheme uses m bits, and that those m bits can be compressed into $b \approx rp\log(m/p)$ bits (see Section 2.5), then our error, by Theorem 2.4.2 is

$$\big|\|c_x - c_y\| - \|x-y\|\big| \lesssim c(m/p)^{-r+1/2} + \varepsilon\|x-y\|,$$

which in rate-distortion terms is

$$\big|\|c_x - c_y\| - \|x-y\|\big| \lesssim 2^{-\frac{b}{p}\frac{r-1/2}{r}} + \varepsilon\|x-y\|.$$

In other words, up to constants in the exponent, and possible logarithmic terms, our result is near-optimal.

2.12 Acknowledgements

The authors would like to thank Sjoerd Dirksen for inspiring discussions and suggestions. Our work was supported in part by NSF Grant DMS-2012546 and a UCSD senate research award. This chapter, in full, is joint work with Rayan Saab and has been published in International Conference on Learning Representations (ICLR), 2021. The dissertation author was the primary investigator and author of this paper.

References

[1] Nir Ailon and Bernard Chazelle. "The fast Johnson–Lindenstrauss transform and approximate nearest neighbors". In: *SIAM Journal on computing* 39.1 (2009), pp. 302–322.

[2] Nir Ailon and Edo Liberty. "An almost optimal unrestricted fast Johnson-Lindenstrauss transform". In: *ACM Transactions on Algorithms (TALG)* 9.3 (2013), pp. 1–12.

[3] Alexandr Andoni and Piotr Indyk. "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions". In: *2006 47th annual IEEE symposium on foundations of computer science (FOCS'06)*. IEEE. 2006, pp. 459–468.

[4] Anna Choromanska, Krzysztof Choromanski, Mariusz Bojarski, Tony Jebara, Sanjiv Kumar, and Yann LeCun. "Binary embeddings with structured hashed projections". In: *International Conference on Machine Learning*. 2016, pp. 344–353.

[5] Evan Chou and C Sinan Güntürk. "Distributed noise-shaping quantization: I. Beta duals of finite frames and near-optimal quantization of random measurements". In: *Constructive Approximation* 44.1 (2016), pp. 1–22.

[6] Kenneth L Clarkson and David P Woodruff. "Low-rank approximation and regression in input sparsity time". In: *Journal of the ACM (JACM)* 63.6 (2017), pp. 1–45.

[7] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. “Locality-sensitive hashing scheme based on p-stable distributions”. In: *Proceedings of the twentieth annual symposium on Computational geometry*. 2004, pp. 253–262.

[8] Ingrid Daubechies and Ron DeVore. “Approximating a bandlimited function using very coarsely quantized data: A family of stable sigma-delta modulators of arbitrary order”. In: *Annals of mathematics* 158.2 (2003), pp. 679–710.

[9] Percy Deift, Felix Krahmer, and C Sınan Güntürk. “An optimal family of exponentially accurate one-bit Sigma-Delta quantization schemes”. In: *Communications on Pure and Applied Mathematics* 64.7 (2011), pp. 883–919.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[11] Sjoerd Dirksen and Alexander Stollenwerk. “Binarized Johnson-Lindenstrauss embeddings”. In: *arXiv preprint arXiv:2009.08320* (2020).

[12] Sjoerd Dirksen and Alexander Stollenwerk. “Fast binary embeddings with Gaussian circulant matrices”. In: *2017 International Conference on Sampling Theory and Applications (SampTA)*. IEEE. 2017, pp. 231–235.

[13] Sjoerd Dirksen and Alexander Stollenwerk. “Fast binary embeddings with gaussian circulant matrices: improved bounds”. In: *Discrete & Computational Geometry* 60.3 (2018), pp. 599–626.

[14] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. “Optimized product quantization for approximate nearest neighbor search”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 2946–2953.

[15] Yunchao Gong, Sanjiv Kumar, Henry A Rowley, and Svetlana Lazebnik. “Learning binary codes for high-dimensional data using bilinear projections”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013, pp. 484–491.

[16] Yunchao Gong, Sanjiv Kumar, Vishal Verma, and Svetlana Lazebnik. “Angular quantization based binary codes for fast similarity search”. In: *Advances in neural information processing systems*. 2012, pp. 1196–1204.

[17] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. “Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval”. In: *IEEE transactions on pattern analysis and machine intelligence* 35.12 (2012), pp. 2916–2929.

[18] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. “Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval”. In: *IEEE transactions on pattern analysis and machine intelligence* 35.12 (2012), pp. 2916–2929.

[19] C Sinan Güntürk. “One-bit sigma-delta quantization with exponential accuracy”. In: *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 56.11 (2003), pp. 1608–1630.

[20] David Harvey and Joris Van Der Hoeven. “Integer multiplication in time O (n log n)”. In: *Preprint* (2019).

[21] Thang Huynh and Rayan Saab. “Fast binary embeddings and quantized compressed sensing with structured matrices”. In: *Communications on Pure and Applied Mathematics* 73.1 (2020), pp. 110–149.

[22] Herve Jegou, Matthijs Douze, and Cordelia Schmid. “Product quantization for nearest neighbor search”. In: *IEEE transactions on pattern analysis and machine intelligence* 33.1 (2010), pp. 117–128.

[23] William B Johnson and Joram Lindenstrauss. “Extensions of Lipschitz mappings into a Hilbert space”. In: *Contemporary mathematics* 26.189-206 (1984), p. 1.

[24] Daniel M Kane and Jelani Nelson. “A derandomized sparse Johnson-Lindenstrauss transform”. In: *arXiv preprint arXiv:1006.3585* (2010).

[25] Daniel M Kane and Jelani Nelson. “Sparser johnson-lindenstrauss transforms”. In: *Journal of the ACM (JACM)* 61.1 (2014), pp. 1–23.

[26] Saehoon Kim, Jungtaek Kim, and Seungjin Choi. “On the Optimal Bit Complexity of Circulant Binary Embedding.” In: *AAAI*. 2018, pp. 3423–3430.

[27] Felix Krahmer and Rachel Ward. “New and improved Johnson–Lindenstrauss embeddings via the restricted isometry property”. In: *SIAM Journal on Mathematical Analysis* 43.3 (2011), pp. 1269–1281.

[28] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. “Cifar-10 (canadian institute for advanced research)”. In: *URL http://www. cs. toronto. edu/kriz/cifar. html* 5 (2010).

[29] Ping Li, Anshumali Shrivastava, Joshua L Moore, and Arnd C König. “Hashing algorithms for large-scale learning”. In: *Advances in neural information processing systems*. 2011, pp. 2672–2680.

[30] Wei Liu, Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. “Hashing with graphs”. In: *ICML*. 2011.

[31] Jiří Matoušek. “On variants of the Johnson–Lindenstrauss lemma”. In: *Random Structures & Algorithms* 33.2 (2008), pp. 142–156.

[32] Jelani Nelson, Eric Price, and Mary Wootters. “New constructions of RIP matrices with fast multiplication and fewer rows”. In: *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*. SIAM. 2014, pp. 1515–1528.

[33] Samet Oymak, Christos Thrampoulidis, and Babak Hassibi. “Near-optimal sample complexity bounds for circulant binary embedding”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pp. 6359–6363.

[34] Bryan A. Plummer, Liwei Wang, Christopher M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. “Flickr30K Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models”. In: *IJCV* 123.1 (2017), pp. 74–93.

[35] Maxim Raginsky and Svetlana Lazebnik. “Locality-sensitive binary codes from shift-invariant kernels”. In: *Advances in neural information processing systems*. 2009, pp. 1509–1517.

[36] Ali Rahimi and Benjamin Recht. “Random features for large-scale kernel machines”. In: *Advances in neural information processing systems* 20 (2007), pp. 1177–1184.

[37] Jorge Sánchez and Florent Perronnin. “High-dimensional signature compression for large-scale image classification”. In: *CVPR 2011*. IEEE. 2011, pp. 1665–1672.

[38] Anshumali Shrivastava and Ping Li. “In defense of minhash over simhash”. In: *Artificial Intelligence and Statistics*. 2014, pp. 886–894.

[39] Yan Xia, Kaiming He, Pushmeet Kohli, and Jian Sun. “Sparse projections for high-dimensional binary codes”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3332–3339.

[40] Xinyang Yi, Constantine Caramanis, and Eric Price. “Binary embedding: Fundamental limits and fast algorithm”. In: *International Conference on Machine Learning*. 2015, pp. 2162–2170.

[41] Felix Yu, Sanjiv Kumar, Yunchao Gong, and Shih-Fu Chang. “Circulant binary embedding”. In: *International conference on machine learning*. 2014, pp. 946–954.

[42] Xu Zhang, Felix X Yu, Ruiqi Guo, Sanjiv Kumar, Shengjin Wang, and Shi-Fu Chang. "Fast orthogonal projection based on kronecker product". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 2929–2937.

Chapter 3

Sigma-Delta and Distributed Noise Shaping Quantization Methods for Random Fourier Features

We propose the use of low bit-depth Sigma-Delta and distributed noise-shaping methods for quantizing the Random Fourier features (RFFs) associated with shift-invariant kernels. We prove that our quantized RFFs – even in the case of 1-bit quantization – allow a high accuracy approximation of the underlying kernels, and the approximation error decays at least polynomially fast as the dimension of the RFFs increases. We also show that the quantized RFFs can be further compressed, yielding an excellent trade-off between memory use and accuracy. Namely, the approximation error now decays exponentially as a function of the bits used. Moreover, we empirically show by testing the performance of our methods on several machine learning tasks that our method compares favorably to other state of the art quantization methods in this context.

3.1 Introduction

Kernel methods have long been demonstrated as effective techniques in various machine learning applications, cf. [33, 32]. Given a dataset $\mathscr{X} \subset \mathbb{R}^d$ with $|\mathscr{X}| = N$, kernel methods *implicitly* map data points to a high, possibly infinite, dimensional feature space $\mathscr{H}$ by $\phi : \mathscr{X} \to \mathscr{H}$. However, instead of working directly on that space the inner products between feature embeddings can be preserved by a kernel function $k(x,y) := \langle \phi(x), \phi(y) \rangle_{\mathscr{H}}$ that coincides

with the inner product. Nevertheless, in cases where N is large, using nonlinear kernels for applications like, say, support vector machines (SVM) and logistic regression requires the expensive computation of the $N \times N$ Gram matrix of the data [24]. In order to overcome this bottleneck, one popular approach is to "linearize" k by using the random Fourier features (RFFs) originally proposed by [29], and in turn built on Bochner's theorem [26]. Given a continuous, shift-invariant real-valued kernel $k(x,y) = \kappa(x-y)$ with $\kappa(0) = 1$, then κ is the (inverse) Fourier transform of a probability measure Λ over $\mathbb{R}^d$ and we have

$$\kappa(u) = \mathbb{E}_{\omega\sim\Lambda} \exp(i\omega^\top u) = \mathbb{E}_{\omega\sim\Lambda} \cos(\omega^\top u). \tag{3.1}$$

As an example, the radial basis function (RBF) kernel $k(x,y) = \exp(-\|x-y\|_2^2/2\sigma^2)$ corresponds to the multivariate normal distribution $\Lambda = \mathcal{N}(0, \sigma^{-2} I_d)$. Following [29], for a target dimension m, the associated RFFs (without normalization) are

$$z(x) := \cos(\Omega^\top x + \xi) \in \mathbb{R}^m \tag{3.2}$$

where $\Omega := (\omega_1, \ldots, \omega_m) \in \mathbb{R}^{d\times m}$ is a random matrix generated as $\omega_j \overset{iid}{\sim} \Lambda$ and $\xi \in \mathbb{R}^m$ is a random vector with $\xi_j \overset{iid}{\sim} U([0, 2\pi))$ for all j. Additionally, the identity $\mathbb{E}(\langle z(x), z(y)\rangle) = \frac{m}{2} k(x,y)$ implies that the inner product of low-dimensional features $\sqrt{\frac{2}{m}} z(x)$, $\sqrt{\frac{2}{m}} z(y)$ can approximate $k(x,y)$ in kernel-based algorithms. Learning a linear model on the (normalized) RFFs then amounts to using the approximation

$$\widehat{k}_{\text{RFF}}(x,y) := \frac{2}{m} \langle z(x), z(y)\rangle \tag{3.3}$$

as a reference kernel during training. For instance, performing linear SVM and linear ridge regression on RFFs winds up training nonlinear kernel-based SVM and ridge regression with $\widehat{k}_{\text{RFF}}$. It turns out that using RFFs in such a way with adjustable dimension m can remarkably speed up training for large-scale data and alleviate the memory burden for storing the kernel matrix. As an

additional and very important benefit, the entire kernel function k is approximated accurately, i.e., the approximation error $|k(x,y) - \widehat{k}_{\text{RFF}}(x,y)|$ has been shown to be small, particularly when m is large, e.g., in [30, 5, 37, 35, 3, 4].

The need for large m for guaranteeing good generalization performance on large datasets [38, 27, 1, 25] provides an opportunity for further savings in memory usage. Rather than store the RFFs in full precision, quantization methods have been proposed to encode RFFs (3.2) into a sequence of bits and subsequently approximate $k(x,y)$ by taking inner product between quantized RFFs, thereby introducing a new level of approximation. One of our goals is to propose quantization techniques that favorably trade off approximation accuracy against number of bits used.

3.1.1 Related Work

To make the discussion more precise, let us start by defining the $2K$-level quantization alphabet that we use throughout as

$$\mathscr{A} = \left\{ \frac{a}{2K-1} \,\middle|\, a = \pm 1, \pm 3, \ldots, \pm(2K-1) \right\}, \tag{3.4}$$

and note that one can use $b := \log_2(2K)$ bits to represent each element of $\mathscr{A}$. The goal of quantization in the RFF context is to map $z(x) = \cos(\Omega^T x + \xi) \in \mathbb{R}^m \mapsto q(x) \in \mathscr{A}^m$. We will be interested in very small values of K, particularly $K = 1$, which corresponds to very few bits per RFF sample.

It is natural to start our discussion of quantization methods with the simplest quantizer, namely memoryless scalar quantization (MSQ), where we round each coordinate of the input vector $z \in \mathbb{R}^m$ to the nearest element in $\mathscr{A}$. Specifically, $Q_{\text{MSQ}} : \mathbb{R}^m \to \mathscr{A}^m$ is defined by

$$q_i := (Q_{\text{MSQ}}(z))_i := \operatorname{argmin}_{v \in \mathscr{A}} |z_i - v|, \quad i = 1, \ldots, m.$$

Moreover, by setting $K = 1$, one can get a binary embedding $Q_{\text{MSQ}}(z) = \text{sign}(z)$ with $\mathscr{A} = \{-1, 1\}$ where sign is an element-wise operation. This yields the so-called one-bit universal quantizer [8, 31] for RFFs, which generates a distorted (biased) kernel

$$\widehat{k}_q(x,y) := \frac{1}{m}\langle \text{sign}(z(x)), \text{sign}(z(y))\rangle. \tag{3.5}$$

Although replacing the sign function in (3.5) by Q_{MSQ} with $K > 1$ and renormalizing the inner product correspondingly can alleviate the distortion, there are better choices in terms of approximation error. In [23], a Lloyd-Max (LM) quantization scheme is designed based on the MSQ where, rather than use the evenly spaced alphabet in (3.4), one has to construct specific alphabets for different K. Recently with an eye towards asymmetric sensor network applications, an asymmetric semi-quantized scheme (SemiQ) was proposed in [31], and shown to be unbiased. It generates $\widehat{k}_s(x,y)$, which is an inner product between an *unquantized* RFF vector and a quantized one, i.e.

$$\widehat{k}_s(x,y) := \frac{\pi}{2m}\langle z(x), Q_{\text{MSQ}}(z(y))\rangle. \tag{3.6}$$

However, this asymmetric setting is restrictive on many kernel machines because it only works for the inference stage and the model still has to be trained based on unquantized RFFs. Another unbiased quantization scheme resorts to injecting randomness into the quantization, and is known as randomized rounding [40], or stochastic quantization (StocQ) [23]. Specifically, for each $z \in \mathbb{R}$, one chooses the two consecutive points $s, t \in \mathscr{A}$ with $z \in [s,t]$. Then one randomly assigns the quantization via $\text{P}(Q_{\text{StocQ}}(z) = s) = \frac{t-z}{t-s}$, $\text{P}(Q_{\text{StocQ}}(z) = t) = \frac{z-s}{t-s}$. It follows that

$$\widehat{k}_{\text{StocQ}}(x,y) := \frac{2}{m}\langle Q_{\text{StocQ}}(z(x)), Q_{\text{StocQ}}(z(y))\rangle \tag{3.7}$$

where Q_{StocQ} operates on each component separately. Due to the Bernoulli sampling for Q_{StocQ}, the quantization process involves additional randomness for each dimension of RFFs, which

leads to extra variance especially in the case of binary embedding, i.e., $b = 1$. Nevertheless, the kernel approximation error for $\widehat{k}_s$ and $\widehat{k}_{\text{StocQ}}$ is bounded by $O(m^{-1/2})$ with high probability, see [31, 40].

3.1.2 Methods and Contributions

We explore the use of ΣΔ [14, 15, 18] and distributed noise-shaping [9, 10] quantization methods on RFFs. These techniques, explicitly defined and discussed in Section 3.2 and Section 3.6, yield superior performance to methods based on scalar quantization in contexts ranging from bandlimited function quantization [14, 18], to quantization of linear measurements [6, 7], of compressed sensing measurements [19], of non-linear measurements [20], and even for binary embeddings that preserve (Euclidean) distances [21, 41]. It is therefore natural to wonder whether they can also yield superior performance in the RFF context. Let $Q_{\Sigma\Delta}^{(r)}$ be the r-th order ΣΔ quantizer and let Q_β be the distributed noise shaping quantizer with $\beta \in (1,2)$, and let $\widetilde{V}_{\Sigma\Delta}$ and $\widetilde{V}_\beta$ be their associated sparse condensation matrices defined in Section 3.2. Then our method approximates kernels via

$$\widehat{k}_{\Sigma\Delta}^{(r)}(x,y) := \langle \widetilde{V}_{\Sigma\Delta} Q_{\Sigma\Delta}^{(r)}(z(x)), \widetilde{V}_{\Sigma\Delta} Q_{\Sigma\Delta}^{(r)}(z(y)) \rangle \tag{3.8}$$

and

$$\widehat{k}_\beta(x,y) := \langle \widetilde{V}_\beta Q_\beta(z(x)), \widetilde{V}_\beta Q_\beta(z(y)) \rangle. \tag{3.9}$$

Specifically, given large-scale data $\mathscr{T}$ contained in a compact set $\mathscr{X} \subset \mathbb{R}^d$, we put forward Algorithm 3 to generate and store quantized RFFs such that one can subsequently use them for training and inference using linear models.

For illustration, Section 3.7 presents a pointwise comparison of above kernel approximations on a synthetic toy dataset. A summary of our contributions follows.

- We give the first detailed analysis of ΣΔ and distributed noise-shaping schemes for quantizing RFFs. Specifically, Theorem 3.3.1 provides a uniform upper bound for the errors

Algorithm 3: Quantized kernel machines

Input: Shift-invariant kernel k, alphabet $\mathscr{A}$, and training data $\mathscr{T} = \{x_i\}_{i=1}^N \subset \mathscr{X}$

Generate random matrix $\Omega \in \mathbb{R}^{d\times m}$ and random vector $\xi \in \mathbb{R}^m$ as in (3.2)

for $i = 1$ **to** N **do**

  $z_i \leftarrow \cos(\Omega^\top x_i + \xi) \in \mathbb{R}^m$ ▷ Compute RFFs

  $q_i \leftarrow Q(z_i) \in \mathscr{A}^m$ ▷ $Q = Q_{\Sigma\Delta}^{(r)}$ or Q_β as in (3.10) and (3.13)

  $y_i \leftarrow \widetilde{V} q_i$ ▷ Further compression with $\widetilde{V} = \widetilde{V}_{\Sigma\Delta}$ or $\widetilde{V}_\beta$ as in (3.14)

Store $\{y_i\}_{i=1}^N$ and use it to train kernel machines with a linear kernel, i.e. inner product

$|\widehat{k}_{\Sigma\Delta}^{(r)}(x,y) - k(x,y)|$ and $|\widehat{k}_\beta(x,y) - k(x,y)|$ over compact (possibly infinite) sets. Our analysis shows that the quantization error decays fast as m grows. Additionally, Theorem 3.3.3 provides spectral approximation guarantees for first order $\Sigma\Delta$ quantized RFF approximation of kernels.

- Our methods allow a further reduction in the number of bits used. Indeed, to implement (3.8) and (3.9) in practice, one would store and transmit the condensed bitstreams $\widetilde{V}_{\Sigma\Delta} Q_{\Sigma\Delta}^{(r)}(z(x))$ or $\widetilde{V}_\beta Q_\beta(z(x))$. For example, since the matrices $\widetilde{V}_{\Sigma\Delta}$ are sparse and essentially populated by bounded integers, each sample can be represented by fewer bits, as summarized in Table 3.1.

- We illustrate the benefits of our proposed methods in several numerical experiments involving kernel ridge regression (KRR), kernel SVM, and two-sample tests based on maximum mean discrepancy (MMD) (all in Section 3.4). Our experiments show that $Q_{\Sigma\Delta}^{(r)}$ and Q_β are comparable with the semi-quantization scheme and outperforms the other fully-quantized method mentioned above, both when we fix the number of RFF features m, and when we fix the number of bits used to store each quantized RFF vector.

3.2 Noise Shaping Quantization Preliminaries

The methods we consider herein are special cases of noise shaping quantization schemes (see, e.g., [11]). For a fixed alphabet $\mathscr{A}$ and each dimension m, such schemes are associated with

an $m \times m$ lower triangular matrix H with unit diagonal, and are given by a map $Q : \mathbb{R}^m \to \mathscr{A}^m$ with $y \mapsto q$ designed to satisfy $y - q = Hu$. The schemes are called *stable* if $\|u\|_\infty \leq C$ where C is independent of m. Among these noise shaping schemes, we will be interested in stable r^{th} order $\Sigma\Delta$ schemes $Q_{\Sigma\Delta}^{(r)}$ [18, 15], and distributed noise shaping schemes Q_β [9, 10]. For example, in the case of $\Sigma\Delta$ with $r = 1$, the entries $q_i, i = 1, ..., m$ of the vector $q = Q_{\Sigma\Delta}^{(1)}(y)$ are assigned iteratively via

$$\begin{cases} u_0 = 0, \\ q_i = Q_{\text{MSQ}}\big(y_i + u_{i-1}\big), \\ u_i = u_{i-1} + y_i - q_i, \end{cases} \tag{3.10}$$

where $Q_{\text{MSQ}}(z) = \text{argmin}_{v \in \mathscr{A}} |z - v|$. This yields the difference equation $y - q = Du$ where D is the first order difference matrix given by $D_{ij} = 1$ if $i = j$, $D_{ij} = -1$ if $i = j + 1$, and 0 otherwise. Stable $\Sigma\Delta$ schemes with $r > 1$, are more complicated to construct (see Section 3.6), but satisfy

$$D^r u = y - q. \tag{3.11}$$

On the other hand, a distributed noise-shaping quantizer $Q_\beta : \mathbb{R}^m \to \mathscr{A}^m$ converts the input vector $y \in \mathbb{R}^m$ to $q = Q_\beta(y) \in \mathscr{A}^m$ such that

$$Hu = y - q \tag{3.12}$$

where, again, $\|u\|_\infty \leq C$. Here, denoting the $p \times p$ identity matrix by I_p and the Kronecker product by $\otimes$, H is a block diagonal matrix defined as $H := I_p \otimes H_\beta \in \mathbb{R}^{m \times m}$ where $H_\beta \in \mathbb{R}^{\lambda \times \lambda}$ is given by $(H_\beta)_{ij} = 1$ if $i = j$, $(H_\beta)_{ij} = -\beta$ if $i = j + 1$, and 0 otherwise. Defining $\widetilde{H} := I_m - H$, one can implement the quantization step $q = Q_\beta(y)$ via the following iterations for $i = 1, 2, \ldots, m$

[9, 10]:

$$\begin{cases} u_0 = 0, \\ q_i = Q_{\mathrm{MSQ}}\big(y_i + \widetilde{H}_{i,i-1}u_{i-1}\big), \\ u_i = y_i + \widetilde{H}_{i,i-1}u_{i-1} - q_i, \end{cases} \tag{3.13}$$

where $Q_{\mathrm{MSQ}}(z) = \mathrm{argmin}_{v\in\mathscr{A}}|z-v|$. The stability of (3.13) is discussed in Section 3.6. It is worth mentioning that since $Q_{\Sigma\Delta}^{(r)}$ and Q_β are sequential quantization methods, they can not be implemented entirely in parallel. On the other hand, blocks of size λ can still be run in parallel. Next, we adopt the definition of a condensation operator in [9, 21, 41].

Definition 3.2.1 ($\Sigma\Delta$ condensation operator)**.** *Let p, r, λ be fixed positive integers such that $\lambda = r\widetilde{\lambda} - r + 1$ for some integer $\widetilde{\lambda}$. Let $m = \lambda p$ and v be a row vector in $\mathbb{R}^\lambda$ whose entry v_j is the j-th coefficient of the polynomial $(1 + z + \ldots + z^{\widetilde{\lambda}-1})^r$. Define the condensation operator $V_{\Sigma\Delta} \in \mathbb{R}^{p\times m}$ as $V_{\Sigma\Delta} := I_p \otimes v$.*

For example, when $r = 1$, $\lambda = \widetilde{\lambda}$ and the vector $v \in \mathbb{R}^\lambda$ is simply the vector of all ones while when $r = 2$, $\lambda = 2\widetilde{\lambda} - 1$ and $v = (1, 2, \ldots, \widetilde{\lambda} - 1, \widetilde{\lambda}, \widetilde{\lambda} - 1, \ldots, 2, 1) \in \mathbb{R}^\lambda$.

Definition 3.2.2 (Distributed noise-shaping condensation operator)**.** *Let p, λ be positive integers and fix $\beta \in (1,2)$. Let $m = \lambda p$ and $v_\beta := (\beta^{-1}, \beta^{-2}, \ldots, \beta^{-\lambda}) \in \mathbb{R}^\lambda$ be a row vector. Define the distributed noise-shaping condensation operator $V_\beta \in \mathbb{R}^{p\times m}$ as $V_\beta := I_p \otimes v_\beta$.*

We will also need the normalized condensation operators given by

$$\widetilde{V}_{\Sigma\Delta} := \frac{\sqrt{2}}{\sqrt{p}\|v\|_2} V_{\Sigma\Delta}, \qquad \widetilde{V}_\beta := \frac{\sqrt{2}}{\sqrt{p}\|v_\beta\|_2} V_\beta. \tag{3.14}$$

If $\widetilde{V}$ is either of the two normalized matrices in (3.14), Lemma 3.9.3 (Section 3.9) shows that

$$\mathbb{E}(\langle \widetilde{V}z(x), \widetilde{V}z(y)\rangle) = k(x,y). \tag{3.15}$$

3.3 Main Results and Space Complexity

Our approach to quantizing RFFs given by (3.8) and (3.9) is justified by (3.15), along with the observation that, for our noise-shaping schemes, we have $q = z - Hu$ with guarantees that $\|\widetilde{V}Hu\|_2$ is small.

Moreover, as we will see in Section 3.3.1, we are able to control the approximation error such that $\widehat{k}_{\Sigma\Delta}(x,y) \approx k(x,y)$ and $\widehat{k}_{\beta}(x,y) \approx k(x,y)$ hold with high probability. In fact Theorem 3.3.1 shows more: the approximation error of the quantized kernel estimators in (3.8) and (3.9) have polynomial and exponential error decay respectively as a function of m, the dimension of the RFFs. Armed with this result, in Section 3.3.2 we also present a brief analysis of the space-complexity associated with our quantized RFFs, and show that the approximation error due to quantization decays exponentially as a function of the bits needed.

Additionally, in various applications such as Kernel Ridge Regression (KRR), spectral error bounds on the kernel may be more pertinent than point-wise bounds. For example, it was shown in [4, 40] that the expected loss of kernel ridge regression performed using an approximation of the true kernel is bounded by a function of the spectral error in the kernel approximation (Lemma 2 of [4], Proposition 1 of [40]). In Theorem 3.3.3, we provide spectral approximation guarantees for first order $\Sigma\Delta$ quantized RFF approximation of kernels, in the spirit of the analogous guarantees in [40] for stochastic quantization.

3.3.1 Approximation error bounds

Point-wise error bounds on the approximation

We begin with Theorem 3.3.1, with its proof in Section 3.9.

Theorem 3.3.1. *Let $\mathscr{X} \subseteq \mathbb{R}^d$ be compact and $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ be a normalized, i.e. $k(0,0) = 1$, shift-invariant kernel. Let Λ be its corresponding probability measure as in* (3.1)*, and suppose that the second moment $\sigma_\Lambda^2 = \mathbb{E}_{\omega\sim\Lambda}\|\omega\|_2^2$ exists. Let $\beta \in (1,2)$, $p, r \in \mathbb{N}$, $\lambda = O(\sqrt{p\log^{-1} p}) \in \mathbb{N}$, and $m = \lambda p$. For $x, y \in \mathscr{X}$, and b-bit alphabet $\mathscr{A}$ in* (3.4) *with $b = \log_2(2K)$, consider the*

approximated kernels $\widehat{k}^{(r)}_{\Sigma\Delta}(x,y)$ *and* $\widehat{k}_{\beta}(x,y)$ *defined as in* (3.8) *and* (3.9) *respectively. Then there exist positive constants* $\{\alpha_i\}_{i=1}^{10}$ *that are independent of* m, p, λ *such that*

$$\sup_{x,y\in\mathscr{X}} \left|\widehat{k}^{(r)}_{\Sigma\Delta}(x,y) - k(x,y)\right| \lesssim \left(\frac{\log p}{p}\right)^{1/2} + \frac{\log^{1/2} p}{\lambda^{r-1}(2^b-1)} + \frac{1}{\lambda^{2r-1}(2^b-1)^2} \tag{3.16}$$

holds with probability at least $1-\alpha_1 p^{-1-\alpha_2} - \alpha_3 \exp(-\alpha_4 p^{1/2} + \alpha_5 \log p)$*, and*

$$\sup_{x,y\in\mathscr{X}} \left|\widehat{k}_{\beta}(x,y) - k(x,y)\right| \lesssim \left(\frac{\log p}{p}\right)^{1/2} + \frac{p^{1/2}}{\beta^{\lambda-1}(2^b-1)} + \frac{1}{\beta^{2\lambda-2}(2^b-1)^2} \tag{3.17}$$

holds with probability exceeding $1-\alpha_6 p^{-1-\alpha_7} - \alpha_8 \exp(-\alpha_9 p^{1/2} + \alpha_{10} \log p)$.

Note that the first error term in (3.16), (3.17) results from the condensation of RFFs, i.e. Theorem 3.9.8, while the remaining two error terms are due to the corresponding quantization schemes.

Spectral approximation guarantees for first order Sigma-Delta quantized RFFs

We begin with a definition of a (Δ_1, Δ_2)-spectral approximation of a matrix as the error bounds Δ_1 and Δ_2 play a key role in bounding the generalization error in various applications such as Kernel Ridge Regression (KRR) (Lemma 2 of [4], Proposition 1 of [40]).

Definition 3.3.2 ((Δ_1,Δ_2)-spectral approximation)**.** *Given* $\Delta_1, \Delta_2 > 0$*, a matrix A is a* (Δ_1,Δ_2)*-spectral approximation of another matrix B if* $(1-\Delta_1)B \preccurlyeq A \preccurlyeq (1+\Delta_2)B$.

For the tractability of obtaining spectral error bounds, in this section we consider a variation of the sigma-delta scheme for $r = 1$. In particular, given a b-bit alphabet as in (3.4) with $b = \log_2(2K)$, we consider the following first-order $\Sigma\Delta$ quantization scheme for a random Fourier feature vector $z(x) \in [-1,1]^m$ corresponding to a data point $x \in \mathbb{R}^d$, where, the state

variable $(u_x)_0$ is initialized as a random number, i.e.

$$\begin{aligned}(u_x)_0 &\sim U\left[-\frac{1}{2^b-1},\frac{1}{2^b-1}\right]\\ q_{i+1} &= Q_{MSQ}((z(x))_{i+1}+(u_x)_i)\\ (u_x)_{i+1} &= (u_x)_i+(z(x))_{i+1}-q_{i+1}\end{aligned} \tag{3.18}$$

where $q \in \mathscr{A}^m$ represents the $\Sigma\Delta$ quantization of $z(x)$ and $(u_x)_0$ is drawn randomly from the uniform distribution on $\left[-\frac{1}{2^b-1},\frac{1}{2^b-1}\right]$.

Let $Q_{\Sigma\Delta}$ be the first order $\Sigma\Delta$ quantizer represented by (3.18) and let $\widetilde{V}_{\Sigma\Delta}$ be the associated sparse condensation matrix as in definition 3.2.1. Then the elements of the corresponding approximation $\hat{K}_{\Sigma\Delta}$ of the kernel K is given by

$$\widehat{K}_{\Sigma\Delta}(x,y) := \langle \widetilde{V}_{\Sigma\Delta}Q_{\Sigma\Delta}(z(x)), \widetilde{V}_{\Sigma\Delta}Q_{\Sigma\Delta}(z(y))\rangle.$$

Now, we state Theorem 3.3.3 whose proof can be found in Section 3.10.

Theorem 3.3.3. *Let $\hat{K}_{\Sigma\Delta}$ be an approximation of a true kernel matrix K using m-feature first-order $\Sigma\Delta$ quantized RFF (as in* (3.18)*) with a b-bit alphabet (as in* (3.4)*) and $m = \lambda p$. Then given $\Delta_1 \geq 0, \Delta_2 \geq \frac{\delta}{\eta}$ where $\eta > 0$ represents the regularization and $\delta = \frac{8+\frac{26}{3p}}{\lambda(2^b-1)^2}$, we have*

$$\begin{aligned}&\mathbb{P}[(1-\Delta_1)(K+\eta I) \preccurlyeq (\hat{K}_{\Sigma\Delta}+\eta I) \preccurlyeq (1+\Delta_2)(K+\eta I)]\\ \geq\ & 1-4n\left[\exp\left(\frac{-p\eta^2\Delta_1^2}{4n\lambda(\frac{1}{\eta}(\|K\|_2+\delta)+2\Delta_1/3)}\right)+\exp\left(\frac{-p\eta^2(\Delta_2-\frac{\delta}{\eta})^2}{4n\lambda(\frac{1}{\eta}(\|K\|_2+\delta)+2(\Delta_2-\frac{\delta}{\eta})/3)}\right)\right].\end{aligned}$$

The above result differs from the spectral bound results presented in [40] for stochastic quantization in a particular aspect of the the lower bound requirement on Δ_2, namely, the lower bound for Δ_2 in Theorem 3.3.3 for first order $\Sigma\Delta$ quantization has another controllable parameter λ in addition to the number of bits b. Specifically, provided $8 >> \frac{26}{3p}$, we have $\delta \approx \frac{8}{\lambda(2^b-1)^2}$, which is monotonically decreasing in λ.

3.3.2 Space complexity

At first glance, Theorem 3.3.1 shows that Q_β has faster quantization error decay as a function of λ (hence m) as compared to $Q_{\Sigma\Delta}^{(r)}$. However, a further compression of the bit-stream resulting from the latter is possible, and results in a similar performance of the two methods from the perspective of bit-rate versus approximation error, as we will now show.

Indeed, our methods entail training and testing linear models on condensed bitstreams $\widetilde{V}q \in \widetilde{V}\mathscr{A}^m \subset \mathbb{R}^p$ where q is the quantized RFFs generated by $Q_{\Sigma\Delta}^{(r)}$ or Q_β, and $\widetilde{V}$ is the corresponding normalized condensation operator. Thus, when considering the space complexity associated with our methods, the relevant factor is the number of bits needed to encode $\widetilde{V}q$. To that end, by storing the normalization factors in $\widetilde{V}$ (see (3.14)) separately using a constant number of bits, we can simply ignore them when considering space complexity. Let us now consider b-bit alphabets $\mathscr{A}$ with $b = \log_2(2K)$. Since the entries of v are integer valued and $\|v\|_1 = O(\lambda^r)$, one can store $\widetilde{V}_{\Sigma\Delta}q$ using $B := O(p\log_2(2K\|v\|_1)) = O(p(b + r\log_2\lambda))$ bits. Then $\lambda^{-r} \approx 2^{-cB/p}$ and thus the dominant error terms in (3.16) decay exponentially as a function of bit-rate B. On the other hand, for distributed noise shaping each coordinate of $\widetilde{V}_\beta q$ is a linear combination of λ components in q, so $\widetilde{V}_\beta q$ takes on at most $(2K)^\lambda$ values. This implies we need $p\log_2(2K)^\lambda = mb$ bits to store $\widetilde{V}_\beta q$ in the worst case.

Remark 3.3.4. *Despite this tight upper bound for arbitrary $\beta \in (1,2)$, an interesting observation is that the number of bits used to store $\widetilde{V}_\beta q$ can be smaller than mb with special choices of β, e.g., when $\beta^k = \beta + 1$ with integer $k > 1$. For example, if $k = 2$ and $b = 1$, then $\beta = (\sqrt{5}+1)/2$ is the* golden ratio *and one can see that $v_\beta = (\beta^{-1}, \ldots, \beta^{-\lambda})$ satisfies $v_\beta(i) = v_\beta(i+1) + v_\beta(i+2)$ for $1 \le i \le \lambda - 2$. Since $b = 1$, we have $q \in \{\pm 1\}^m$ and $\widetilde{V}_\beta q$ (ignoring the normalizer) can be represented by $p\log_2(\beta^\lambda) = m\log_2(\beta) < m$ bits. Defining the number of bits used to encode each RFF vector by $R := m\log_2(\beta)$, then* (3.17) *shows that $\beta^{-\lambda} = 2^{-\lambda R/m} = 2^{-R/p}$ dominates the error. In other words, up to constants, the error is essentially equal to the error obtained by a λ bit MSQ quantization of a p-dimensional RFF embedding.*

If we assume that each full-precision RFF is represented by 32 bits, then the storage cost per sample for both full-precision RFF and semi-quantized scheme Q_{SemiQ} in (3.6) is $32m$. Because Q_{StocQ} in (3.7) does not admit further compression, it needs mb bits. A comparison of space complexity of different methods is summarized in Table 3.1.

Table 3.1. The memory usage to store each encoded sample.

Method	RFFs	Q_{SemiQ}	Q_{StocQ}	$Q_{\Sigma\Delta}^{(r)}$	Q_β
Memory	$32m$	$32m$	mb	$O(p(b+r\log_2\lambda))$	mb^*

* This can be reduced to $mb\log_2\beta$ for certain β.

3.4 Numerical Experiments

We have established that both $Q_{\Sigma\Delta}^{(r)}$ and Q_β are memory efficient and approximate their intended kernels well. In this section, we will verify via numerical experiments that they perform favorably compared to other baselines on machine learning tasks.

3.4.1 Kernel Ridge Regression

Kernel ridge regression (KRR) [28] corresponds to the ridge regression (linear least squares with ℓ_2 regularization) in a reproducing kernel Hilbert space (RKHS). We synthesize $N = 5000$ highly nonlinear data samples $(x_i, y_i) \in \mathbb{R}^5 \times \mathbb{R}$ such that for each i, we draw each component of $x_i \in \mathbb{R}^5$ uniformly from $[-1,1)$ and use it to generate

$$y_i = f(x_i) := \gamma_1^\top x_i + \gamma_2^\top \cos(x_i^2) + \gamma_3^\top \cos(|x_i|) + \varepsilon_i$$

where $\gamma_1 = \gamma_2 = \gamma_3 = [1,1,\ldots,1]^\top \in \mathbb{R}^5$, and $\varepsilon_i \sim \mathcal{N}(0, \frac{1}{4})$. This is split into 4000 samples used for training and 1000 samples for testing. Given a RBF kernel $k(x,y) = \exp(-\gamma\|x-y\|_2^2)$ with $\gamma = 1/d = 0.2$, by the representer theorem, our predictor is of the form $\widehat{f}(x) = \sum_{i=1}^N \alpha_i k(x_i, x)$ where the coefficient vector $\alpha := (\alpha_1, \ldots, \alpha_N) \in \mathbb{R}^N$ is obtained by solving $(K + \eta I_N)\alpha = y$. Here, $K = (k(x_i, x_j)) \in \mathbb{R}^{N\times N}$ is the kernel matrix and $\eta = 1$ is the regularization parameter.

Since the dimension of RFFs satisfies $m = \lambda p$, there is a trade-off between p and λ. According to Theorem 3.3.1, increasing the embedding dimension p can reduce the error caused by compressing RFFs, while larger λ leads to smaller quantization error and makes the memory usage of $Q_{\Sigma\Delta}^{(r)}$ more efficient (see Table 3.1). Beyond this, all hyperparameters, e.g. λ, β, are tuned based on cross validation. In our experiment, we consider the kernel approximations $\widehat{k}_{\text{RFF}}$, $\widehat{k}_{\text{StocQ}}$, $\widehat{k}_{\Sigma\Delta}^{(1)}$ with $\lambda = 15$, $\widehat{k}_{\Sigma\Delta}^{(2)}$ with $\lambda = 15$, and $\widehat{k}_{\beta}$ with $\beta = 1.9$, $\lambda = 12$. These are applied for both training (solving for α) and testing (computing $\widehat{f}(x)$ based on α), while the semi-quantized scheme $\widehat{k}_s$ is only used for testing and its coefficient vector α is learned by using $\widehat{k}_{\text{RFF}}$ on the training set. Furthermore, according to [31], $\widehat{k}_s$ can be used in two scenarios during the testing stage:

1. Training data is unquantized RFFs while test data is quantized, i.e., $\widehat{f}(x) = \sum_{i=1}^{N} \alpha_i \widehat{k}_s(x_i, x)$;
2. Quantize training data and leave testing points as RFFs, i.e., $\widehat{f}(x) = \sum_{i=1}^{N} \alpha_i \widehat{k}_s(x, x_i)$.

We summarize the KRR results averaging over 30 runs for $b = 1$ bit quantizers in Figure 3.1, in which solid curves represent our methods and the dashed lines depict other baselines. Note that in both cases, the noise-shaping quantizer Q_β achieves the lowest test mean squared error (MSE) among all quantization schemes, and it even outperforms the semi-quantization scheme $\widehat{k}_s$ with respect to the number of measurements m. Moreover, due to the further compression advantage, $Q_{\Sigma\Delta}^{(r)}$ and Q_β are more memory efficient than the fully-quantized scheme Q_{StocQ} in terms of the usage of bits per sample. More experiments for $b = 2, 3$ can be found in Section 3.8.

3.4.2 Kernel SVM

To illustrate the performance of our methods for classification tasks, we perform Kernel SVM [32, 36] to evaluate different kernel approximations on the UCI ML hand-written digits dataset [2, 39], in which $N = 1797$ grayscale images compose $C = 10$ classes and they are vectorized to $d = 64$ dimensional vectors. Additionally, all pixel values are scaled in the range

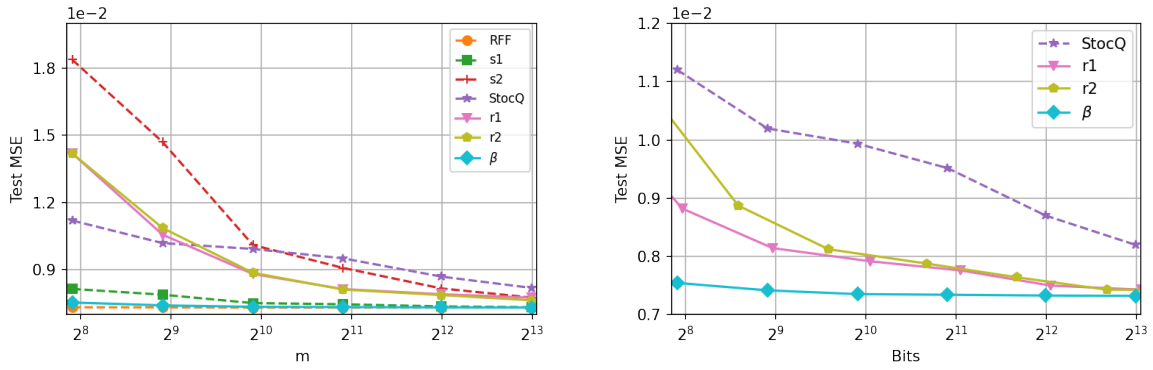


Figure 3.1. Kernel ridge regression with $b = 1$. The labels RFF, $s1$, $s2$, StocQ, $r1$, $r2$, β represent $\widehat{k}_{\mathrm{RFF}}$, $\widehat{k}_s$ for scenarios (1), (2), $\widehat{k}_{\mathrm{StocQ}}$, $\widehat{k}^{(1)}_{\Sigma\Delta}$, $\widehat{k}^{(2)}_{\Sigma\Delta}$, and $\widehat{k}_\beta$ respectively.

$[0,1]$ and we randomly split this dataset into 80% for training and 20% for testing. As for the classifier, we use the soft margin SVM with a regularization parameter $R = 1$.

Note that in the binary classification case, i.e. labels $y_i \in \{-1,1\}$, our goal is to learn the coefficients α_i, the intercept b, and the index set of support vectors S in a decision function during the training stage:

$$g(x) := \mathrm{sign}\Big(\sum_{i\in \mathrm{S}} \alpha_i y_i k(x,x_i) + b\Big). \tag{3.19}$$

Here, we use a RBF kernel $k(x,y) = \exp(-\gamma\|x-y\|_2^2)$ with $\gamma = 1/(d\sigma_0^2) \approx 0.11$ and σ_0^2 being equal to the variance of training data. In the multi-class case, we implement the "one-versus-one" approach for multi-class classification where $\frac{C(C-1)}{2}$ classifiers are constructed and each one trains data from two classes. In our experiment, we found that a large embedding dimension $p = m/\lambda$ is needed and approximations $\widehat{k}_{\mathrm{RFF}}$, $\widehat{k}_{\mathrm{StocQ}}$, $\widehat{k}^{(1)}_{\Sigma\Delta}$ with $\lambda = 2$, $\widehat{k}^{(2)}_{\Sigma\Delta}$ with $\lambda = 3$, and $\widehat{k}_\beta$ with $\beta = 1.1$, $\lambda = 2$, are implemented for both training (obtaining α_i, b, and S in (3.19)) and testing (predicting the class of an incoming sample x by $g(x)$) phases, whereas the asymmetric scheme $\widehat{k}_s$ is only performed for inference with its parameters in (3.19) learned from $\widehat{k}_{\mathrm{RFF}}$ during the training stage. Moreover, as before there are two versions of $\widehat{k}_s$ used for making predictions:

1. Keep the support vectors as unquantized RFFs and quantize the test point x, i.e. substitute $\widehat{k}_s(x_i,x)$ for $k(x,x_i)$ in (3.19);

2. Quantize the support vectors and leave the testing point x as unquantized RFFs, i.e., replace $k(x,x_i)$ in (3.19) with $\widehat{k}_s(x,x_i)$.

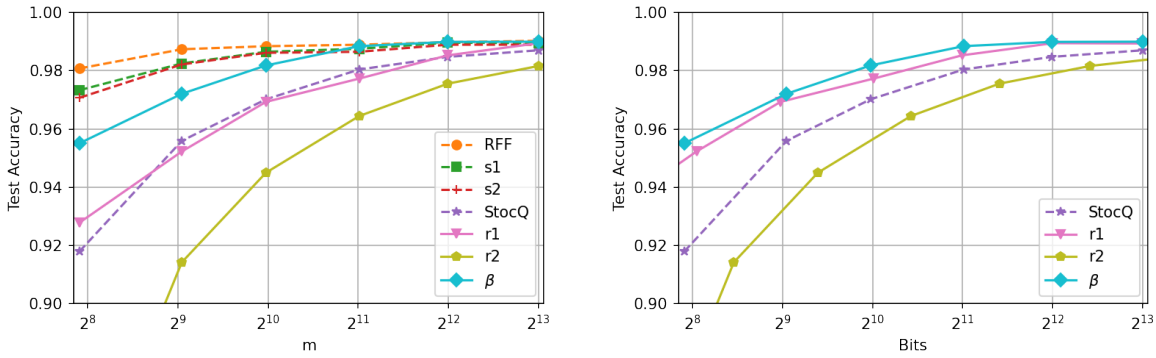


Figure 3.2. Kernel SVM with $b = 1$. The labels RFF, $s1$, $s2$, StocQ, $r1$, $r2$, β represent $\widehat{k}_{\text{RFF}}$, $\widehat{k}_s$ for scenarios (1), (2), $\widehat{k}_{\text{StocQ}}$, $\widehat{k}^{(1)}_{\Sigma\Delta}$, $\widehat{k}^{(2)}_{\Sigma\Delta}$, and $\widehat{k}_\beta$ respectively.

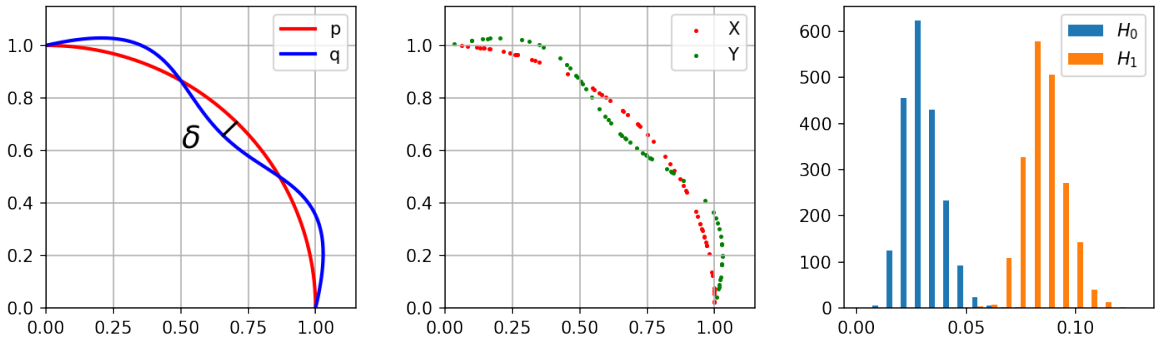
For each binary quantization scheme (with $b = 1$), the average test accuracy over 30 independent runs is plotted in Figure 3.2. We observe that, in regard to m, Q_β substantially outperforms other fully-quantized schemes including $Q^{(r)}_{\Sigma\Delta}$ and Q_{StocQ}, but, as expected, it is still worse than the semi-quantized methods. Memory efficiency is characterized in the right plot by estimating the test accuracy against the storage cost (in terms of bits) per sample. Note that both Q_β and $Q^{(1)}_{\Sigma\Delta}$ have significant advantage over the baseline method Q_{StocQ}, which means that our methods require less memory to achieve the same test accuracy when $b = 1$. See Section 3.8 for extra experiment results with $b = 2, 3$.

3.4.3 Maximum Mean Discrepancy

Given two distributions p and q, and a kernel k over $\mathscr{X} \subset \mathbb{R}^d$, the maximum mean discrepancy (MMD) has been shown to play an important role in the *two-sample test* [17], by proposing the null hypothesis $\mathscr{H}_0 : p = q$ against the alternative hypothesis $\mathscr{H}_1 : p \neq q$. The square of MMD distance can be computed by

$$\text{MMD}^2_k(p,q) = \mathbb{E}_{x,x'}(k(x,x')) + \mathbb{E}_{y,y'}(k(y,y')) - 2\mathbb{E}_{x,y}(k(x,y))$$

where $x, x' \overset{iid}{\sim} p$ and $y, y' \overset{iid}{\sim} q$. Here, we set k to a RBF kernel, which is *characteristic* [34] implying that $\mathrm{MMD}_k(p,q)$ is metric, i.e. $\mathrm{MMD}_k(p,q) = 0 \iff p = q$, and the following hypothesis test is consistent.



(a) True distributions: p and q **(b)** Samples of size 60 from p, q **(c)** Counts vs MMD values

Figure 3.3. Two distributions and the MMD values based on the RBF kernel.

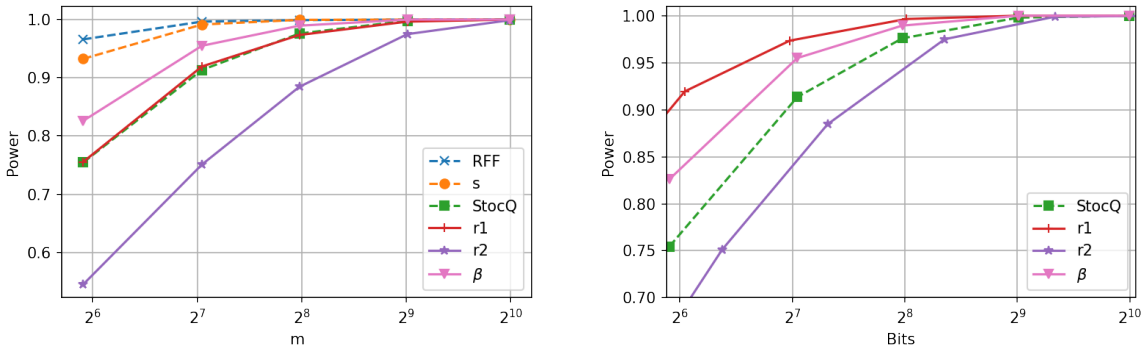


Figure 3.4. Power of the permutation test with $b = 1$. The labels RFF, s, StocQ, $r1$, $r2$, β represent $\widehat{k}_{\mathrm{RFF}}$, $\widehat{k}_s$, $\widehat{k}_{\mathrm{StocQ}}$, $\widehat{k}^{(1)}_{\Sigma\Delta}$, $\widehat{k}^{(2)}_{\Sigma\Delta}$, and $\widehat{k}_\beta$ respectively.

In our experiment, the distribution p is supported on a quadrant of the unit circle while q is generated by perturbing p by a gap of size δ at various regions, see Figure 3.3a. Let $n = 60$ and choose finite samples $X = \{x_1, \ldots, x_n\} \sim p$ and $Y = \{y_1, \ldots, y_n\} \sim q$. Then $\mathrm{MMD}_k(p,q)$ can be estimated by

$$\widehat{\mathrm{MMD}}^2_k(X,Y) := \frac{1}{n^2}\sum_{i,j=1}^{n} k(x_i,x_j) + \frac{1}{n^2}\sum_{i,j=1}^{n} k(y_i,y_j) - \frac{2}{n^2}\sum_{i,j=1}^{n} k(x_i,y_j). \tag{3.20}$$

Under the null hypothesis $\mathscr{H}_0$, one can get the empirical distribution of (3.20) by reshuffling the data samples $X \cup Y$ many times ($t = 2000$) and recomputing $\widehat{\mathrm{MMD}}_k^2(X', Y')$ on each partition $X' \cup Y'$. For a significance level of $\alpha = 0.05$, $\mathscr{H}_0$ is rejected if the original $\widehat{\mathrm{MMD}}_k^2(X, Y)$ is greater than the $(1-\alpha)$ quantile from the empirical distribution. Figure 3.3c shows that the empirical distributions of (3.20) under both $\mathscr{H}_0$ and $\mathscr{H}_1$ are separated well, where we use the ground truth RBF kernel with small bandwidth $\sigma = 0.05$.

In order to compare different quantization methods when $b = 1$, we use the following approximations with optimal λ to perform the permutation test: $\widehat{k}_{\mathrm{RFF}}$, $\widehat{k}_{\mathrm{StocQ}}$, $\widehat{k}^{(1)}_{\Sigma\Delta}$ with $\lambda = 4$, $\widehat{k}^{(2)}_{\Sigma\Delta}$ with $\lambda = 5$, and $\widehat{k}_\beta$ with $\beta = 1.5$, $\lambda = 4$. Due to the symmetry in (3.20), $\widehat{k}_s$ can be implemented without worrying about the order of inputs. Additionally, if the probability of Type II error, i.e. false negative rate, is denoted by β, then the statistical power of our test is defined by

$$\text{power} = 1 - \beta = \mathrm{P}(\text{reject}\, \mathscr{H}_0 \,|\, \mathscr{H}_1 \text{is true})$$

In other words, the power equals to the portion of MMD values under $\mathscr{H}_1$ that are greater than the $(1-\alpha)$ quantile of MMD distribution under $\mathscr{H}_0$. In Figure 3.4, we observe that, compared with other fully-quantized schemes, Q_β has the greatest power in terms of m. The performance of semi-quantized scheme is pretty close to the plain RFF approximation while it requires more storage space, as discussed in Section 3.3. Moreover, Figure 3.5 presents the corresponding changes of the MMD distributions under $\mathscr{H}_0$ and $\mathscr{H}_1$, in which the overlap between the two distributions is considerably reduced as m increases. Regarding the number of bits per sample, both $Q^{(1)}_{\Sigma\Delta}$ and Q_β have remarkable advantage over Q_{StocQ}. Extra results related to $b = 2, 3$ can be found in Section 3.8.

3.5 Conclusion

In order to reduce memory requirement for training and storing kernel machines, we proposed a framework of using Sigma-Delta and distributed noise-shaping quantization schemes,

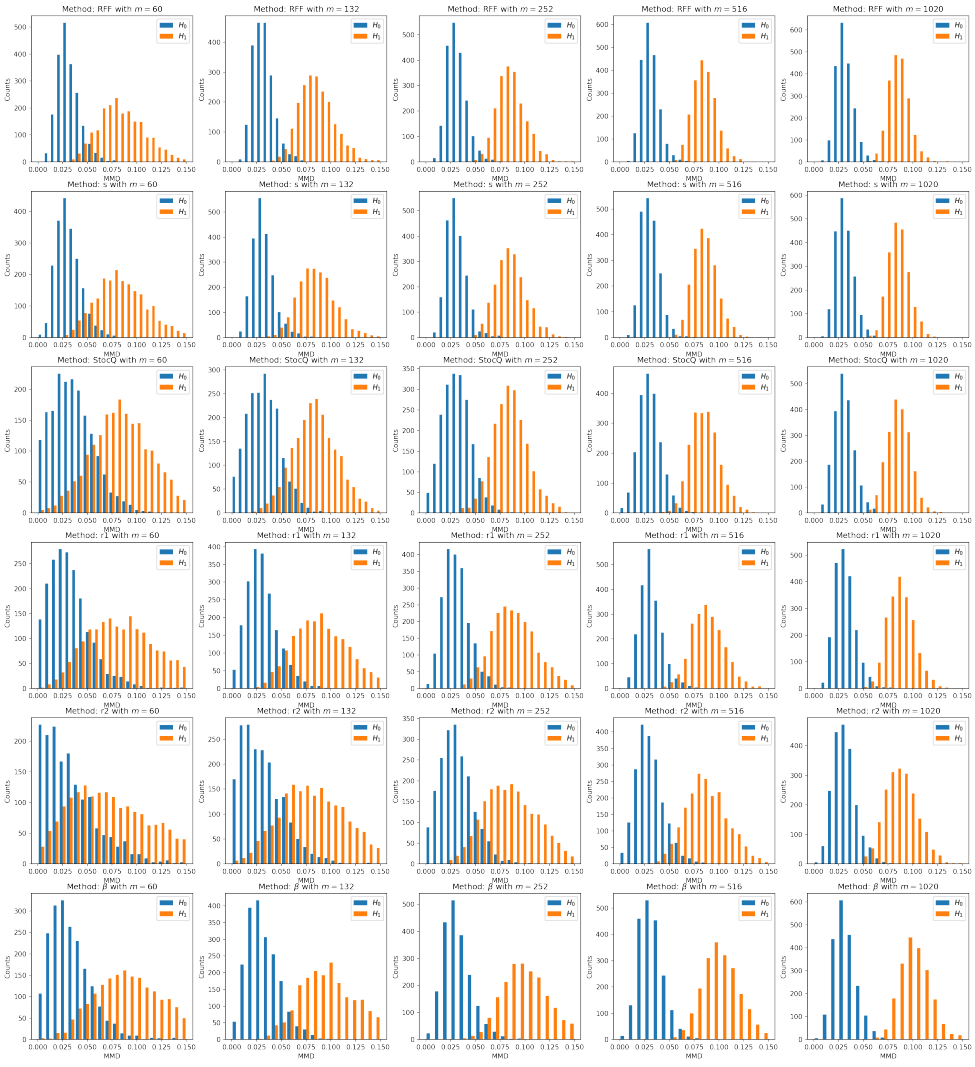


Figure 3.5. The empirical distributions of MMD values under $\mathscr{H}_0$ and $\mathscr{H}_1$.

$Q_{\Sigma\Delta}^{(r)}$ and Q_β, to approximate shift-invariant kernels. We have shown that these fully deterministic quantization schemes are capable of saving more bits than other baselines without compromising the performance. Importantly, we showed that, for all pairs of signals from an infinite low-complexity set, the approximations have uniform probabilistic error bounds yielding an exponential decay as the number of bits used increases. Empirically, we illustrated across popular kernel machines that the proposed quantization methods achieve strong performance both as a function of the dimension of the RFF embedding, and the number of bits used, especially in the case of binary embedding.

3.6 Stable Quantization Methods

The general definition for stable $Q_{\Sigma\Delta}^{(r)}$. Although it is a non-trivial task to design a stable $Q_{\Sigma\Delta}^{(r)}$ for $r > 1$, families of $\Sigma\Delta$ quantization schemes that achieve this goal have been designed [14, 15, 18], and we adopt the version in [15]. Specifically, an r-th order $\Sigma\Delta$ quantization scheme may also arise from the following difference equation

$$y - q = H * v \tag{3.21}$$

where $*$ is the convolution operator and the sequence $H := D^r g$ with $g \in \ell^1$. Then any bounded solution v of (3.21) gives rise to a bounded solution u of (3.11) via $u = g * v$. By change of variables, (3.11) can be reformulated as (3.21). By choosing a proper filter $h := \delta^{(0)} - H$, where $\delta^{(0)}$ denotes the Kronecker delta sequence supported at 0, one can implement (3.21) by $v_i = (h*v)_i + y_i - q_i$ and the corresponding stable quantization scheme $Q_{\Sigma\Delta}^{(r)}$ reads as

$$\begin{cases} q_i = Q((h*v)_i + y_i), \\ v_i = (h*v)_i + y_i - q_i. \end{cases} \tag{3.22}$$

Furthermore, the above design leads to the following result from [15, 22], which exploits the constant $c(K,\mu,r)$ to bound $\|u\|_\infty$.

Proposition 3.6.1. *There exists a universal constant $C > 0$ such that the $\Sigma\Delta$ schemes* (3.10) *and* (3.22) *with alphabet $\mathscr{A}$ in* (3.4)*, are stable, and*

$$\|y\|_\infty \le \mu < 1 \Longrightarrow \|u\|_\infty \le c(K,r) := \frac{CC_1^r r^r}{2K-1},$$

where $C_1 = \left(\left\lceil \frac{\pi^2}{(\cosh^{-1}\gamma)^2} \right\rceil \frac{e}{\pi}\right)$ with $\gamma := 2K - (2K-1)\mu$.

Note that even with the $b = 1$ bit alphabet, i.e. $K = 1$ and $\mathscr{A} = \{-1, 1\}$, stability can be

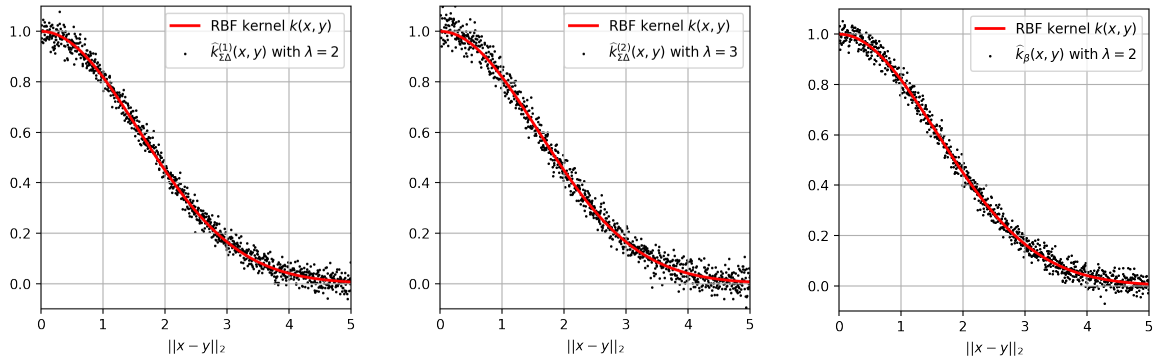
guaranteed with

$$\|y\|_\infty \le \mu < 1 \quad \Longrightarrow \quad \|u\|_\infty \le C \cdot C_1^r \cdot r^r.$$

The stability of Q_β. The relevant result for stability of the noise-shaping quantization schemes (3.13) is the following proposition, which can be simply proved by induction or can be found in [10].

Proposition 3.6.2. *The noise-shaping scheme* (3.13) *with alphabet* $\mathscr{A}$ *in* (3.4) *is stable and*

$$\|y\|_\infty \le \frac{2K-\beta}{2K-1} \Longrightarrow \|u\|_\infty \le c(K,\beta) := \frac{1}{2K-1}.$$



(a) $\widehat{k}^{(r)}_{\Sigma\Delta}(x,y)$ with $r=1$ **(b)** $\widehat{k}^{(r)}_{\Sigma\Delta}(x,y)$ with $r=2$ **(c)** $\widehat{k}_\beta(x,y)$ with $\beta = 1.1$

Figure 3.6. Kernel Approximations with $b = 3$.

3.7 A comparison of kernel approximations

In Figure 3.6, we evaluate approximated kernels (3.8) and (3.9) in Section 3.1 on $n = 1000$ pairs of points $\{x_i, y_i\}_{i=1}^n$ in $\mathbb{R}^d$ with $d = 50$ such that for each i

$$x_i \sim \mathscr{N}(0, I_d), \quad u_i \sim \mathscr{N}(0, I_d), \quad y_i = x_i + \frac{5i}{n} \cdot \frac{u_i}{\|u_i\|_2}.$$

Moreover, each data point x_i is represented by $m = 3000$ RFF features and we use 3-bit quantizers to guarantee good performance for all methods. The target RBF kernel (red curves) is $k(x,y) = \exp(-\|x-y\|_2^2/2\sigma^2)$ with $\gamma := 1/2\sigma^2 = \frac{1}{5}$ and note that the approximations (black dots) have their ℓ_2 distances $\|x-y\|_2$ uniformly distributed in the range $[0,5]$. We see that both $\widehat{k}^{(r)}_{\Sigma\Delta}$ and $\widehat{k}_{\beta}$ can approximate k well.

3.8 More Figures in Section 3.4

KRR. In Figure 3.7 and Figure 3.8, we show the KRR results for $b = 2$ and $b = 3$ respectively. Same as in the Section 3.4, we see that the proposed methods $Q^{(r)}_{\Sigma\Delta}$ and Q_{β} have strong performance in terms of m and the number of bits used for each sample.

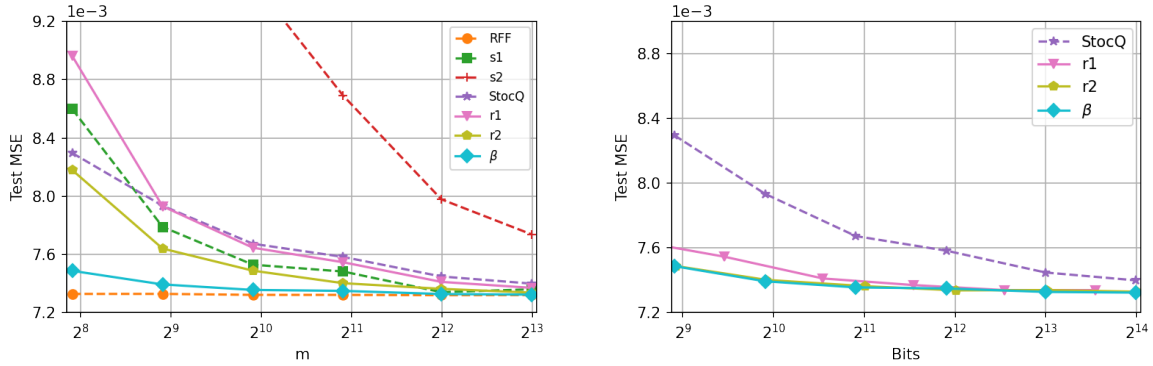


Figure 3.7. Kernel ridge regression with $b = 2$.

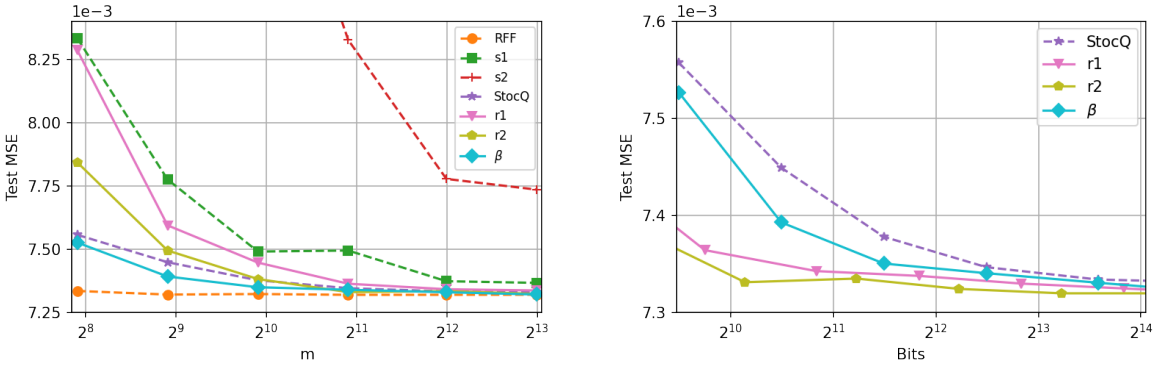


Figure 3.8. Kernel ridge regression with $b = 3$.

Kernel SVM. Figure 3.9 and Figure 3.10 illustrate the performance of kernel SVM with $b = 2$ and $b = 3$ respectively. As we expect, the gap across various schemes shrinks when we use multibit quantizers, where Q_{StocQ} is comparable with $Q^{(r)}_{\Sigma\Delta}$ and Q_{β}.

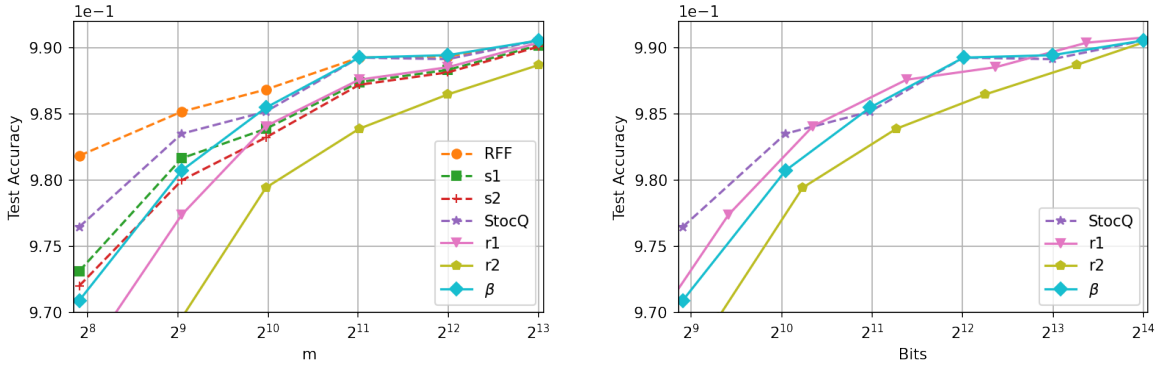


Figure 3.9. Kernel SVM with $b = 2$.

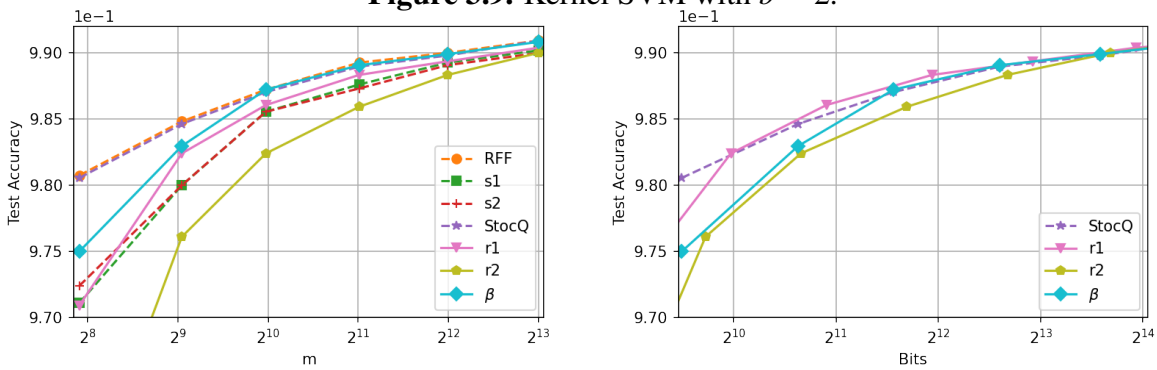


Figure 3.10. Kernel SVM with $b = 3$.

MMD. As a continuation of the two-sample test in Section 3.4, Figure 3.11 and Figure 3.12 imply that both semi-quantized scheme and Q_{StocQ} have better performance with respect to m, while $Q_{\Sigma\Delta}^{(r)}$ can save more memory than other quantization methods.

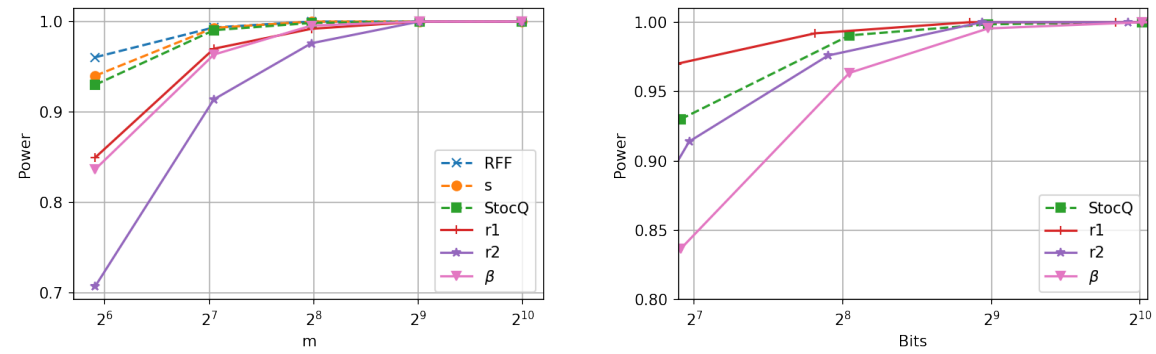


Figure 3.11. Power of the permutation test with $b = 2$.

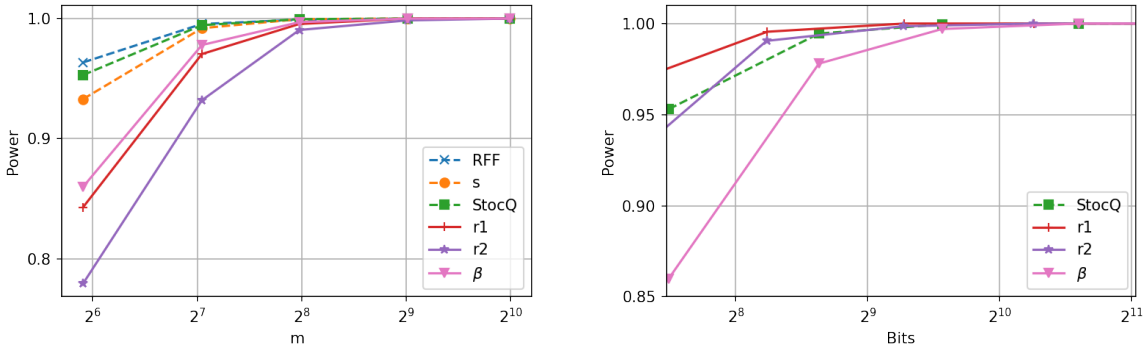


Figure 3.12. Power of the permutation test with $b = 3$.

3.9 Proof of Theorem 3.3.1

Given $x, y \in \mathscr{X} \subset \mathbb{R}^d$, we use either stable $Q_{\Sigma\Delta}^{(r)}$ or stable Q_β to quantize their RFFs $z(x)$, $z(y)$ as in (3.2). Then we get quantized sequences

$$q_{\Sigma\Delta}^{(r)}(x) := Q_{\Sigma\Delta}^{(r)}(z(x)), \quad q_{\Sigma\Delta}^{(r)}(y) := Q_{\Sigma\Delta}^{(r)}(z(y)), \quad \text{or} \quad q_\beta(x) := Q_\beta(z(x)), \quad q_\beta(y) := Q_\beta(z(y)),$$

and expect that both

$$\widehat{k}_{\Sigma\Delta}^{(r)}(x,y) = \langle \widetilde{V}_{\Sigma\Delta} q_{\Sigma\Delta}^{(r)}(x), \widetilde{V}_{\Sigma\Delta} q_{\Sigma\Delta}^{(r)}(y) \rangle, \quad \widehat{k}_\beta(x,y) = \langle \widetilde{V}_\beta q_\beta(x), \widetilde{V}_\beta q_\beta(y) \rangle$$

approximate the ground truth $k(x,y)$ well.

In the case of $\Sigma\Delta$ quantization, by (3.11), one can get

$$\widetilde{V}_{\Sigma\Delta} q_{\Sigma\Delta}^{(r)}(x) = \widetilde{V}_{\Sigma\Delta} z(x) - \widetilde{V}_{\Sigma\Delta} D^r u_x, \quad \widetilde{V}_{\Sigma\Delta} q_{\Sigma\Delta}^{(r)}(y) = \widetilde{V}_{\Sigma\Delta} z(y) - \widetilde{V}_{\Sigma\Delta} D^r u_y.$$

It follows that

$$\begin{aligned}\widehat{k}_{\Sigma\Delta}^{(r)}(x,y) = \langle \widetilde{V}_{\Sigma\Delta} q_{\Sigma\Delta}^{(r)}(x), \widetilde{V}_{\Sigma\Delta} q_{\Sigma\Delta}^{(r)}(y) \rangle &= \langle \widetilde{V}_{\Sigma\Delta} z(x), \widetilde{V}_{\Sigma\Delta} z(y) \rangle - \langle \widetilde{V}_{\Sigma\Delta} z(x), \widetilde{V}_{\Sigma\Delta} D^r u_y \rangle \\ &\quad - \langle \widetilde{V}_{\Sigma\Delta} z(y), \widetilde{V}_{\Sigma\Delta} D^r u_x \rangle + \langle \widetilde{V}_{\Sigma\Delta} D^r u_x, \widetilde{V}_{\Sigma\Delta} D^r u_y \rangle.\end{aligned}$$

The triangle inequality implies that

$$\begin{aligned}\left|\widehat{k}^{(r)}_{\Sigma\Delta}(x,y)-k(x,y)\right| \leq \underbrace{\left|\langle \widetilde{V}_{\Sigma\Delta}z(x), \widetilde{V}_{\Sigma\Delta}z(y)\rangle - k(x,y)\right|}_{\text{(I)}} + \underbrace{\left|\langle \widetilde{V}_{\Sigma\Delta}z(x), \widetilde{V}_{\Sigma\Delta}D^r u_y\rangle\right|}_{\text{(II)}} \\ + \underbrace{\left|\langle \widetilde{V}_{\Sigma\Delta}z(y), \widetilde{V}_{\Sigma\Delta}D^r u_x\rangle\right|}_{\text{(III)}} + \underbrace{\left|\langle \widetilde{V}_{\Sigma\Delta}D^r u_x, \widetilde{V}_{\Sigma\Delta}D^r u_y\rangle\right|}_{\text{(IV)}}.\end{aligned} \tag{3.23}$$

Similarly, for the noise-shaping quantization, one can derive the following inequality based on (3.12),

$$\begin{aligned}\left|\widehat{k}^{(r)}_{\beta}(x,y)-k(x,y)\right| \leq \underbrace{\left|\langle \widetilde{V}_{\beta}z(x), \widetilde{V}_{\beta}z(y)\rangle - k(x,y)\right|}_{\text{(I)}} + \underbrace{\left|\langle \widetilde{V}_{\beta}z(x), \widetilde{V}_{\beta}Hu_y\rangle\right|}_{\text{(II)}} \\ + \underbrace{\left|\langle \widetilde{V}_{\beta}z(y), \widetilde{V}_{\beta}Hu_x\rangle\right|}_{\text{(III)}} + \underbrace{\left|\langle \widetilde{V}_{\beta}Hu_x, \widetilde{V}_{\beta}Hu_y\rangle\right|}_{\text{(IV)}}.\end{aligned} \tag{3.24}$$

In order to control the kernel approximation errors in (3.23) and (3.24), we need to bound fours terms (I), (II), (III), and (IV) on the right hand side.

3.9.1 Useful Lemmata

In this section, we present the following well-known concentration inequalities and relevant lemmata.

Theorem 3.9.1 (Hoeffding's inequality [16]). *Let $X_1,\ldots,X_M$ be a sequence of independent random variables such that $\mathbb{E}X_l = 0$ and $|X_l| \leq B_l$ almost surely for all $1 \leq l \leq M$. Then for all $t > 0$,*

$$\mathrm{P}\Big(\Big|\sum_{l=1}^{M} X_l\Big| \geq t\Big) \leq 2\exp\Big(-\frac{t^2}{2\sum_{l=1}^{M} B_l^2}\Big).$$

Theorem 3.9.2 (Bernstein's inequality [16]). *Let $X_1,\ldots,X_M$ be independent random variables with zero mean such that $|X_l| \leq K$ almost surely for all $1 \leq l \leq M$ and some constant $K > 0$.*

Furthermore assume $\mathbb{E}|X_l|^2 \leq \sigma_l^2$ *for constants* $\sigma_l > 0$ *for all* $1 \leq l \leq M$. *Then for all* $t > 0$,

$$\mathrm{P}\Big(\Big|\sum_{l=1}^{M} X_l\Big| \geq t\Big) \leq 2\exp\Big(-\frac{t^2/2}{\sigma^2 + Kt/3}\Big)$$

where $\sigma^2 := \sum_{l=1}^{M} \sigma_l^2$.

Additionally, one can compute the moments of $\cos(\omega_i^\top x + \xi_i)\cos(\omega_j^\top y + \xi_j)$ as follows.

Lemma 3.9.3. *Suppose* $x, y \in \mathbb{R}^d$ *with RFFs* $z(x)$ *and* $z(y)$ *as in* (3.2). *Let* $\widetilde{V}$ *be either of the two normalized condensation operators defined in* (3.14). *Then*

$$\mathbb{E}(\cos(\omega_j^\top x + \xi_j)\cos(\omega_j^\top y + \xi_j)) = \frac{1}{2}k(x,y), \quad j = 1,2,\ldots,m. \tag{3.25}$$

$$\mathbb{E}(\cos^2(\omega_i^\top x + \xi_i)\cos^2(\omega_j^\top y + \xi_j)) = \begin{cases} \frac{1}{4} + \frac{1}{8}k(2x,2y) & \text{if } i = j, \\ \frac{1}{4} & \text{if } i \neq j. \end{cases} \tag{3.26}$$

$$\mathbb{E}(\langle \widetilde{V}z(x), \widetilde{V}z(y)\rangle) = k(x,y). \tag{3.27}$$

Proof. (i) Using trigonometric identities, the independence of ω_j and ξ_j and formula (3.1), we get

$$\mathbb{E}(\cos(\omega_j^\top x + \xi_j)\cos(\omega_j^\top y + \xi_j)) = \frac{1}{2}\mathbb{E}_{\omega_j \sim \Lambda}\cos(\omega_j^\top (x-y)) = \frac{1}{2}\kappa(x-y) = \frac{1}{2}k(x,y).$$

(ii) If $i = j$, then

$$\begin{aligned} \mathbb{E}(\cos^2(\omega_i^\top x + \xi_i)\cos^2(\omega_i^\top y + \xi_i)) &= \frac{1}{4}\mathbb{E}\Big((1 + \cos(2\omega_i^\top x + 2\xi_i))(1 + \cos(2\omega_i^\top y + 2\xi_i))\Big) \\ &= \frac{1}{4}\Big(1 + \mathbb{E}(\cos(2\omega_i^\top x + 2\xi_i)\cos(2\omega_i^\top y + 2\xi_i))\Big) \\ &= \frac{1}{4} + \frac{1}{8}k(2x,2y). \end{aligned}$$

Similarly, when $i \neq j$ we have

$$\begin{aligned}\mathbb{E}(\cos^2(\omega_i^\top x+\xi_i)\cos^2(\omega_j^\top y+\xi_j)) &= \frac{1}{4}+\frac{1}{4}\mathbb{E}\Big(\cos(2\omega_i^\top x+2\xi_i)\cos(2\omega_j^\top y+2\xi_j)\Big)\\ &= \frac{1}{4}+\frac{1}{4}\mathbb{E}(\cos(2\omega_i^\top x+2\xi_i))\mathbb{E}(\cos(2\omega_j^\top y+2\xi_j))\\ &= \frac{1}{4}.\end{aligned}$$

(iii) According to (3.25), we have $\mathbb{E}(z(x)z(y)^\top) = \frac{1}{2}k(x,y)I_m$ and thus

$$\begin{aligned}\mathbb{E}(\langle \widetilde{V}z(x), \widetilde{V}z(y)\rangle) &= \mathbb{E}(\mathrm{tr}(z(y)^\top \widetilde{V}^\top \widetilde{V}z(x)))\\ &= \mathbb{E}(\mathrm{tr}(\widetilde{V}^\top \widetilde{V}z(x)z(y)^\top))\\ &= \mathrm{tr}(\widetilde{V}^\top \widetilde{V}\mathbb{E}(z(x)z(y)^\top))\\ &= \frac{1}{2}k(x,y)\|\widetilde{V}\|_F^2\\ &= k(x,y).\end{aligned}$$

□

Lemma 3.9.4. *Let $x, y \in \mathbb{R}^d$ and $\varepsilon > 0$. Then*

$$\mathrm{P}\Big(\big|\langle \widetilde{V}_{\Sigma\Delta}z(x), \widetilde{V}_{\Sigma\Delta}z(y)\rangle - k(x,y)\big| \geq \varepsilon\Big) \leq 2\exp\Big(-\frac{\varepsilon^2 p}{2+k(2x,2y)+2k^2(x,y)+(4\lambda+2)\varepsilon/3}\Big),$$

$$\mathrm{P}\Big(\big|\langle \widetilde{V}_{\beta}z(x), \widetilde{V}_{\beta}z(y)\rangle - k(x,y)\big| \geq \varepsilon\Big) \leq 2\exp\Big(-\frac{\varepsilon^2 p}{2+k(2x,2y)+2k^2(x,y)+(4\lambda+2)\varepsilon/3}\Big).$$

Proof. (i) We first consider the case of $\Sigma\Delta$ scheme, i.e. (I) in (3.23). Note that

$$\langle \widetilde{V}_{\Sigma\Delta}z(x), \widetilde{V}_{\Sigma\Delta}z(y)\rangle = \frac{2}{p\|v\|_2^2}\sum_{i=1}^{p} S_i(x,y)$$

where $S_1(x,y),\dots,S_p(x,y)$ are i.i.d. with

$$S_i(x,y) := \sum_{j,k=1}^{\lambda} v_j v_k \cos(\omega_{(i-1)\lambda+j}^\top x + \xi_{(i-1)\lambda+j}) \cos(\omega_{(i-1)\lambda+k}^\top y + \xi_{(i-1)\lambda+k}).$$

Due to (3.25), (3.26), and

$$\mathbb{E}\Big(S_i(x,y) - \frac{k(x,y)\|v\|_2^2}{2}\Big) = 0,$$

one can get

$$\begin{aligned}
\mathrm{Var}\Big(S_i(x,y) - \frac{k(x,y)\|v\|_2^2}{2}\Big) &= \mathrm{Var}(S_i(x,y)) \\
&= \frac{1}{8}\Big(2(k^2(x,y)+1)\|v\|_2^4 + (k(2x,2y) - 4k^2(x,y)) \sum_{i=1}^{\lambda} v_i^4\Big) \\
&\le \frac{\|v\|_2^4}{8}\Big(2k^2(x,y) + 2 + k(2x,2y)\Big)
\end{aligned}$$

and

$$\Big|S_i(x,y) - \frac{k(x,y)\|v\|_2^2}{2}\Big| \le |S_i(x,y)| + \frac{\|v\|_2^2}{2} \le \|v\|_1^2 + \frac{\|v\|_2^2}{2} \le (\lambda + 1/2)\|v\|_2^2$$

for all $1 \le i \le p$, it follows immediately from Bernstein's inequality that

$$\begin{aligned}
\mathrm{P}\Big(\big|\langle \widetilde{V}_{\Sigma\Delta} z(x), \widetilde{V}_{\Sigma\Delta} z(y)\rangle - k(x,y)\big| \ge \varepsilon\Big) &= \mathrm{P}\Big(\Big|\sum_{i=1}^{p}\Big(S_i(x,y) - \frac{k(x,y)\|v\|_2^2}{2}\Big)\Big| \ge \frac{\varepsilon p \|v\|_2^2}{2}\Big) \\
&\le 2\exp\Big(-\frac{\varepsilon^2 p}{2 + k(2x,2y) + 2k^2(x,y) + (4\lambda+2)\varepsilon/3}\Big).
\end{aligned}$$

(ii) Since the proof in part (i) works for all vectors $v \in \mathbb{R}^\lambda$ with nonnegative components, a similar result holds for the noise-shaping case by replacing $V_{\Sigma\Delta}$ and v by V_β and v_β respectively. □

Lemma 3.9.5. *Let $x \in \mathbb{R}^d$ and $\varepsilon > 0$. Then*

$$\mathrm{P}\Big(\frac{1}{p\|v\|_2^2}\|V_{\Sigma\Delta} z(x)\|_1 \ge \varepsilon\Big) \le 2p\exp\Big(-\frac{\varepsilon^2\|v\|_2^2}{1 + 2\varepsilon\|v\|_\infty/3}\Big),$$

$$\mathrm{P}\Big(\frac{1}{p\|v\|_2^2}\|V_\beta z(x)\|_1 \geq \varepsilon\Big) \leq 2p\exp\Big(-\frac{\varepsilon^2\|v_\beta\|_2^2}{1+2\varepsilon\|v_\beta\|_\infty/3}\Big).$$

Proof. (i) In the case of $\Sigma\Delta$ quantization, we note that $V_{\Sigma\Delta} = I_p \otimes v$ and

$$\frac{1}{\|v\|_2^2}V_{\Sigma\Delta}z(x) = \frac{(I_p\otimes v)z(x)}{\|v\|_2^2} = \begin{bmatrix} R_1(x) \\ \vdots \\ R_p(x) \end{bmatrix}$$

where $R_i(x) := \frac{1}{\|v\|_2^2}\sum_{j=1}^{\lambda} v_j z(x)_{(i-1)\lambda+j} = \frac{1}{\|v\|_2^2}\sum_{j=1}^{\lambda} v_j \cos(\omega_{(i-1)\lambda+j}^\top x + \xi_{(i-1)\lambda+j})$ for $1 \leq i \leq p$.

Since $\mathbb{E}(v_j^2\cos^2(\omega_{(i-1)\lambda+j}^\top x+\xi_{(i-1)\lambda+j})) = v_j^2/2$ and $|v_j\cos(\omega_{(i-1)\lambda+j}^\top x+\xi_{(i-1)\lambda+j})| \leq \|v\|_\infty$ holds for all i and j, we can apply Theorem 3.9.2 to $R_i(x)$ with $K = \|v\|_\infty$, $M = \lambda$, and $\sigma^2 = \frac{\|v\|_2^2}{2}$. Specifically, for all $t > 0$, we have

$$\mathrm{P}(|R_i(x)| \geq t) \leq 2\exp\Big(-\frac{t^2\|v\|_2^2}{1+2t\|v\|_\infty/3}\Big). \tag{3.28}$$

Since

$$\frac{1}{p\|v\|_2^2}\|V_{\Sigma\Delta}z(x)\|_1 = \frac{1}{p}\|\frac{1}{\|v\|_2^2}V_{\Sigma\Delta}z(x)\|_1 = \frac{1}{p}\sum_{i=1}^{p}|R_i(x)|,$$

by union bound, we have

$$\begin{aligned}
\mathrm{P}\Big(\frac{1}{p\|v\|_2^2}\|V_{\Sigma\Delta}z(x)\|_1 \geq \varepsilon\Big) &= \mathrm{P}\Big(\sum_{i=1}^{p}|R_i(x)| \geq \varepsilon p\Big) \\
&\leq \mathrm{P}\Big(\bigcup_{i=1}^{p}\{|R_i(x)| \geq \varepsilon\}\Big) \\
&\leq \sum_{i=1}^{p}\mathrm{P}(|R_i(x)| \geq \varepsilon) \\
&\leq 2p\exp\Big(-\frac{\varepsilon^2\|v\|_2^2}{1+2\varepsilon\|v\|_\infty/3}\Big)
\end{aligned}$$

where the last inequality is due to (3.28).

(ii) Substituting $V_{\Sigma\Delta}$ with $V_\beta = I_p \otimes v_\beta$ leads to a verbatim proof for the second inequality. □

3.9.2 Upper bound of (I)

This section is devoted to deriving an upper bound of the term (I) in (3.23), (3.24). Here, we adapt the proof techniques used in [37].

According to Theorem 3.3.1, $\mathscr{X}$ is a compact subset of $\mathbb{R}^d$ with *diameter* $\ell = \mathrm{diam}(\mathscr{X}) > 0$. Then $\mathscr{X}^2 := \mathscr{X} \times \mathscr{X}$ is a compact set in $\mathbb{R}^{2d}$ with diameter $\sqrt{2}\ell$. Additionally, the second moment of distribution Λ, that is, $\sigma_\Lambda^2 := \mathbb{E}_{\omega\sim\Lambda}\|\omega\|_2^2 = \mathrm{tr}(\nabla^2\kappa(0))$ exists where $\nabla^2\kappa$ is the Hessian matrix of κ in (3.1). We will need the following results in order to obtain a uniform bound of term (I) over $\mathscr{X}$, via an ε-net argument.

Lemma 3.9.6 ([12])**.** *Let $B_2^d(\eta) := \{x \in \mathbb{R}^d : \|x\|_2 \le \eta\}$. Then the covering number $\mathscr{N}(B_2^d(\eta), \varepsilon)$ satisfies*

$$\mathscr{N}(B_2^d(\eta), \varepsilon) \le \Big(\frac{4\eta}{\varepsilon}\Big)^d.$$

Lemma 3.9.7 (Jung's Theorem [13])**.** *Let $K \subseteq \mathbb{R}^d$ be compact with $\mathrm{diam}(K) > 0$. Then K is contained in a closed ball with radius*

$$\eta \le \mathrm{diam}(K)\sqrt{\frac{d}{2(d+1)}}.$$

The boundary case of equality is attained by the regular n-simplex.

We can now prove the following theorem controlling term (I).

Theorem 3.9.8. *Let $\varepsilon, \eta_1 > 0$. Then*

$$\begin{aligned}
&\mathrm{P}\Big(\sup_{x,y\in\mathscr{X}} |\langle \widetilde{V}_{\Sigma\Delta} z(x), \widetilde{V}_{\Sigma\Delta} z(y)\rangle - k(x,y)| < \varepsilon\Big) \\
&\ge 1 - 32\sigma_\Lambda^2\Big(\frac{\eta_1\lambda}{\varepsilon}\Big)^2 - 2\Big(\frac{4\ell}{\eta_1}\Big)^{2d}\exp\Big(-\frac{\varepsilon^2 p}{8 + 4k(2x,2y) + 8k^2(x,y) + (8\lambda+4)\varepsilon/3}\Big),
\end{aligned}$$

and

$$\begin{aligned}
&\mathrm{P}\Big(\sup_{x,y\in\mathscr{X}}|\langle\widetilde{V}_\beta z(x),\widetilde{V}_\beta z(y)\rangle - k(x,y)|<\varepsilon\Big)\\
&\geq 1-32\sigma_\Lambda^2\Big(\frac{\eta_1\lambda}{\varepsilon}\Big)^2-2\Big(\frac{4\ell}{\eta_1}\Big)^{2d}\exp\Big(-\frac{\varepsilon^2 p}{8+4k(2x,2y)+8k^2(x,y)+(8\lambda+4)\varepsilon/3}\Big).
\end{aligned}$$

Proof. Indeed, the following proof techniques are independent of the choice of row vector v in $\widetilde{V}_{\Sigma\Delta}$. So we only prove the case related to $\widetilde{V}_{\Sigma\Delta}$ and everything works for $\widetilde{V}_\beta$ by replacing v with v_β. Let

$$s(x,y):=\langle\widetilde{V}_{\Sigma\Delta}z(x),\widetilde{V}_{\Sigma\Delta}z(y)\rangle,\quad f(x,y):=s(x,y)-k(x,y).$$

Recall that $\mathbb{E}(s(x,y))=k(x,y)$ and $s(x,y)=\frac{2}{p\|v\|_2^2}\sum_{i=1}^p S_i(x,y)$ where $S_1(x,y),\ldots,S_p(x,y)$ are i.i.d. with

$$S_i(x,y)=\sum_{j,k=1}^{\lambda}v_jv_k\cos(\omega_{(i-1)\lambda+j}^\top x+\xi_{(i-1)\lambda+j})\cos(\omega_{(i-1)\lambda+k}^\top y+\xi_{(i-1)\lambda+k}).$$

According to Lemma 3.9.7, $\mathscr{X}^2\subseteq\mathbb{R}^{2d}$ is enclosed in a closed ball with radius $\ell\sqrt{\frac{2d}{2d+1}}$. By Lemma 3.9.6, one can cover $\mathscr{X}^2$ using an η_1-net with at most $\Big(\frac{4\ell}{\eta_1}\sqrt{\frac{2d}{2d+1}}\Big)^{2d}\leq T_1:=\big(\frac{4\ell}{\eta_1}\big)^{2d}$ balls of radius η_1. Let $c_i=(x_i,y_i)$ denote their centers for $1\leq i\leq T_1$.

For $1\leq l\leq d$ we have

$$\begin{aligned}
&\Big|\frac{\partial s}{\partial x_l}(x,y)\Big|=\frac{2}{p\|v\|_2^2}\Big|\sum_{i=1}^p\frac{\partial S_i}{\partial x_l}(x,y)\Big|\\
&\leq\frac{2}{p\|v\|_2^2}\sum_{i=1}^p\Big|\sum_{j,k=1}^{\lambda}v_jv_k\sin(\omega_{(i-1)\lambda+j}^\top x+\xi_{(i-1)\lambda+j})\cos(\omega_{(i-1)\lambda+k}^\top y+\xi_{(i-1)\lambda+k})\omega_{(i-1)\lambda+j,l}\Big|\\
&\leq\frac{2}{p\|v\|_2^2}\sum_{i=1}^p\sum_{j,k=1}^{\lambda}v_jv_k|\omega_{(i-1)\lambda+j,l}|.
\end{aligned}$$

Then

$$\mathbb{E}\Big(\frac{2}{p\|v\|_2^2}\sum_{i=1}^{p}\sum_{j,k=1}^{\lambda} v_j v_k|\omega_{(i-1)\lambda+j,l}|\Big) \le \frac{2}{p\|v\|_2^2}\sum_{i=1}^{p}\sum_{j,k=1}^{\lambda} v_j v_k \mathbb{E}(\|\omega\|_\infty) = \frac{2\|v\|_1^2}{\|v\|_2^2}E(\|\omega\|_\infty) < \infty.$$

Since $\left|\frac{\partial s}{\partial x_l}(x,y)\right|$ is dominated by an integrable function, one can interchange expectations and partial derivatives. In particular,

$$\frac{\partial}{\partial x_l}\Big(\mathbb{E}s(x,y)\Big) = \mathbb{E}\Big(\frac{\partial s}{\partial x_l}(x,y)\Big),$$

and similarly

$$\frac{\partial}{\partial y_l}\Big(\mathbb{E}s(x,y)\Big) = \mathbb{E}\Big(\frac{\partial s}{\partial y_l}(x,y)\Big).$$

It follows that

$$\mathbb{E}\nabla s(x,y) = \nabla \mathbb{E}s(x,y) = \nabla k(x,y). \tag{3.29}$$

Let $L_f = \|\nabla f(x^*,y^*)\|_2$ be the Lipschitz constant with $(x^*,y^*) = \text{argmax}_{(x,y)\in\mathscr{X}^2}\|\nabla f(x,y)\|_2$. Applying law of total expectation and (3.29) gives

$$\begin{aligned}
\mathbb{E}(L_f^2) &= \mathbb{E}(\|\nabla s(x^*,y^*) - \nabla k(x^*,y^*)\|_2^2) \\
&= \mathbb{E}\big(\mathbb{E}(\|\nabla s(x^*,y^*) - \nabla k(x^*,y^*)\|_2^2 \,|\, x^*,y^*)\big) \\
&= \mathbb{E}\Big(\mathbb{E}(\|\nabla s(x^*,y^*)\|_2^2 \,|\, x^*,y^*) + \|\nabla k(x^*,y^*)\|_2^2 - 2\mathbb{E}(\langle \nabla s(x^*,y^*), \nabla k(x^*,y^*)\rangle \,|\, x^*,y^*)\Big) \\
&= \mathbb{E}(\|\nabla s(x^*,y^*)\|_2^2) + \mathbb{E}\Big(\|\nabla k(x^*,y^*)\|_2^2 - 2\langle \mathbb{E}(\nabla s(x^*,y^*) \,|\, x^*,y^*), \nabla k(x^*,y^*)\rangle\Big) \\
&= \mathbb{E}(\|\nabla s(x^*,y^*)\|_2^2) + \mathbb{E}\Big(\|\nabla k(x^*,y^*)\|_2^2 - 2\langle \nabla k(x^*,y^*), \nabla k(x^*,y^*)\rangle\Big) \\
&= \mathbb{E}(\|\nabla s(x^*,y^*)\|_2^2) - \mathbb{E}(\|\nabla k(x^*,y^*)\|_2^2) \\
&\le \mathbb{E}(\|\nabla s(x^*,y^*)\|_2^2) \\
&= \mathbb{E}(\|\nabla_x s(x^*,y^*)\|_2^2) + \mathbb{E}(\|\nabla_y s(x^*,y^*)\|_2^2).
\end{aligned} \tag{3.30}$$

Note that

$$
\begin{aligned}
&\|\nabla_x s(x^*,y^*)\|_2 \le \frac{2}{p\|v\|_2^2}\sum_{i=1}^{p}\|\nabla_x S_i(x^*,y^*)\|_2 \\
&= \frac{2}{p\|v\|_2^2}\sum_{i=1}^{p}\Big\|\sum_{j,k=1}^{\lambda} v_j v_k \sin(\omega_{(i-1)\lambda+j}^\top x^* + \xi_{(i-1)\lambda+j})\cos(\omega_{(i-1)\lambda+k}^\top y^* + \xi_{(i-1)\lambda+k})\omega_{(i-1)\lambda+j}\Big\|_2 \\
&= \frac{2}{p\|v\|_2^2}\sum_{i=1}^{p}\Big\|\sum_{k=1}^{\lambda} v_k \cos(\omega_{(i-1)\lambda+k}^\top y^* + \xi_{(i-1)\lambda+k}) \\
&\times \sum_{j=1}^{\lambda} v_j \sin(\omega_{(i-1)\lambda+j}^\top x^* + \xi_{(i-1)\lambda+j})\omega_{(i-1)\lambda+j}\Big\|_2 \\
&= \frac{2}{p\|v\|_2^2}\sum_{i=1}^{p}\Big|\sum_{k=1}^{\lambda} v_k \cos(\omega_{(i-1)\lambda+k}^\top y^* + \xi_{(i-1)\lambda+k})\Big| \\
&\times \Big\|\sum_{j=1}^{\lambda} v_j \sin(\omega_{(i-1)\lambda+j}^\top x^* + \xi_{(i-1)\lambda+j})\omega_{(i-1)\lambda+j}\Big\|_2 \\
&\le \frac{2\|v\|_1}{p\|v\|_2^2}\sum_{i=1}^{p}\sum_{j=1}^{\lambda} v_j\|\omega_{(i-1)\lambda+j}\|_2.
\end{aligned}
$$

By Cauchy–Schwarz inequality and the fact $\|v\|_1 \le \sqrt{\lambda}\|v\|_2$, we have

$$
\begin{aligned}
\|\nabla_x s(x^*,y^*)\|_2^2 &\le \frac{4\|v\|_1^2}{p^2\|v\|_2^4}\Big(\sum_{i=1}^{p}\sum_{j=1}^{\lambda} v_j\|\omega_{(i-1)\lambda+j}\|_2\Big)^2 \\
&\le \frac{4\|v\|_1^2}{p\|v\|_2^2}\sum_{i=1}^{p}\sum_{j=1}^{\lambda}\|\omega_{(i-1)\lambda+j}\|_2^2 \\
&\le \frac{4\lambda}{p}\sum_{i=1}^{p}\sum_{j=1}^{\lambda}\|\omega_{(i-1)\lambda+j}\|_2^2.
\end{aligned}
$$

Then

$$
\mathbb{E}(\|\nabla_x s(x^*,y^*)\|_2^2) \le 4\lambda^2\mathbb{E}(\|\omega\|_2^2) = 4\lambda^2\sigma_\Lambda^2
$$

and similarly

$$
\mathbb{E}(\|\nabla_y s(x^*,y^*)\|_2^2) \le 4\lambda^2\sigma_\Lambda^2.
$$

Plugging above results into (3.30) shows

$$\mathbb{E}(L_f^2) \le 8\lambda^2 \sigma_\Lambda^2.$$

Let $\varepsilon > 0$. Markov's inequality implies

$$\mathrm{P}\Big(L_f \ge \frac{\varepsilon}{2\eta_1}\Big) \le \frac{4\eta_1^2 \mathbb{E}(L_f^2)}{\varepsilon^2} \le 32\sigma_\Lambda^2 \Big(\frac{\eta_1 \lambda}{\varepsilon}\Big)^2. \tag{3.31}$$

By union bound and Lemma 3.9.4, we get

$$\begin{aligned}\mathrm{P}\Big(\bigcup_{i=1}^{T_1} \{|f(c_i)| \ge \varepsilon/2\}\Big) &\le \sum_{i=1}^{T_1} \mathrm{P}\Big(|f(c_i)| \ge \varepsilon/2\Big) \\ &\le 2\Big(\frac{4\ell}{\eta_1}\Big)^{2d} \exp\Big(-\frac{\varepsilon^2 p}{8 + 4k(2x,2y) + 8k^2(x,y) + (8\lambda+4)\varepsilon/3}\Big). \end{aligned} \tag{3.32}$$

If $|f(c_i)| < \varepsilon/2$ for all i and $L_f < \varepsilon/2\eta_1$, then $|f(x,y)| < \varepsilon$ for all $(x,y) \in \mathscr{X}^2$. It follows immediately from (3.31) and (3.32) that

$$\begin{aligned}&\mathrm{P}\Big(\sup_{x,y\in\mathscr{X}} |f(x,y)| < \varepsilon\Big) \\ &\ge 1 - 32\sigma_\Lambda^2 \Big(\frac{\eta_1\lambda}{\varepsilon}\Big)^2 - 2\Big(\frac{4\ell}{\eta_1}\Big)^{2d} \exp\Big(-\frac{\varepsilon^2 p}{8 + 4k(2x,2y) + 8k^2(x,y) + (8\lambda+4)\varepsilon/3}\Big).\end{aligned}$$

□

3.9.3 Upper bound of (II) & (III)

By symmetry it suffices to bound (II) in (3.23), (3.24), and the same upper bound holds for (III).

Theorem 3.9.9. *Let $\varepsilon, \eta_2 > 0$. Then we have*

$$\begin{aligned}
&\mathrm{P}\Big(\sup_{x,y\in\mathscr{X}} \big|\langle \widetilde{V}_{\Sigma\Delta} z(x), \widetilde{V}_{\Sigma\Delta} D^r u_y\rangle\big| < \varepsilon \Big) \\
&\geq 1 - \sigma_\Lambda^2 \Big(\frac{c(K,r) 2^{r+2} \eta_2}{\varepsilon} \Big)^2 - 2p \Big(\frac{2\sqrt{2}\ell}{\eta_2} \Big)^d \exp\Big(- \frac{\varepsilon^2 \|v\|_2^2}{c(K,r)^2 2^{2r+4} + c(K,r) 2^{r+3} \varepsilon \|v\|_\infty / 3} \Big).
\end{aligned}$$

and

$$\begin{aligned}
&\mathrm{P}\Big(\sup_{x,y\in\mathscr{X}} \big|\langle \widetilde{V}_{\beta} z(x), \widetilde{V}_{\beta} H u_y\rangle\big| < \varepsilon \Big) \\
&\geq 1 - \sigma_\Lambda^2 \Big(\frac{4\eta_2 c(K,\beta)}{\varepsilon \beta^\lambda} \Big)^2 - 2p \Big(\frac{2\sqrt{2}\ell}{\eta_2} \Big)^d \exp\Big(- \frac{\varepsilon^2 \|v_\beta\|_2^2 \beta^{2\lambda}}{16 c(K,\beta)^2 + 8\beta^\lambda c(K,\beta) \varepsilon \|v_\beta\|_\infty / 3} \Big),
\end{aligned}$$

where $c(K,r)$ and $c(K,\beta)$ are upper bounds of the ℓ_∞ norm of state vectors in Proposition 3.6.1 and Proposition 3.6.2 respectively.

Proof. (i) We first prove the case associated with $\widetilde{V}_{\Sigma\Delta}$. Since $\mathrm{diam}(\mathscr{X}) = \ell$, by Lemma 3.9.6 and Lemma 3.9.7, one can cover $\mathscr{X}$ using an η_2-net with at most $T_2 := \big(\frac{2\sqrt{2}\ell}{\eta_2}\big)^d$ balls with radius η_2. Let x_k denote their centers for $1 \leq k \leq T_2$. For $x \in \mathbb{R}^d$, define

$$g(x) := \frac{1}{p\|v\|_2^2} \|V_{\Sigma\Delta} z(x)\|_1 = \frac{1}{p\|v\|_2^2} \sum_{i=1}^p |g_i(x)|$$

where $g_i(x) := \sum_{j=1}^\lambda v_j \cos(\omega_{(i-1)\lambda+j}^\top x + \xi_{(i-1)\lambda+j})$. By triangle inequality, we have

$$|g(x) - g(y)| \leq \frac{1}{p\|v\|_2^2} \sum_{i=1}^p |g_i(x) - g_i(y)| \leq \Big(\frac{1}{p\|v\|_2^2} \sum_{i=1}^p \|\nabla g_i(x_i^*)\|_2 \Big) \|x - y\|_2 = L_g \|x - y\|_2$$

where $x_i^* = \operatorname{argmax}_{x\in\mathscr{X}} \|\nabla g_i(x)\|_2$ and $L_g := \frac{1}{p\|v\|_2^2}\sum_{i=1}^p \|\nabla g_i(x_i^*)\|_2$. It follows that

$$\begin{aligned} L_g &= \frac{1}{p\|v\|_2^2}\sum_{i=1}^{p}\Big\|\sum_{j=1}^{\lambda} v_j \sin(\omega_{(i-1)\lambda+j}^\top x_i^* + \xi_{(i-1)\lambda+j})\omega_{(i-1)\lambda+j}\Big\|_2 \\ &\le \frac{1}{p\|v\|_2^2}\sum_{i=1}^{p}\sum_{j=1}^{\lambda} v_j\|\omega_{(i-1)\lambda+j}\|_2. \end{aligned}$$

Applying Cauchy-Schwarz inequality gives

$$L_g^2 \le \frac{1}{p^2\|v\|_2^4}\Big(\sum_{i=1}^{p}\sum_{j=1}^{\lambda} v_j^2\Big)\Big(\sum_{i=1}^{p}\sum_{j=1}^{\lambda}\|\omega_{(i-1)\lambda+j}\|_2^2\Big) = \frac{1}{p\|v\|_2^2}\sum_{i=1}^{p}\sum_{j=1}^{\lambda}\|\omega_{(i-1)\lambda+j}\|_2^2.$$

Taking expectation on both sides leads to

$$\mathbb{E}(L_g^2) \le \frac{\lambda}{\|v\|_2^2}\mathbb{E}(\|\omega\|_2^2) \le \mathbb{E}(\|\omega\|_2^2) = \sigma_\Lambda^2.$$

Let $\varepsilon > 0$. Markov's inequality implies

$$\mathrm{P}\Big(L_g \ge \frac{\varepsilon}{2\eta_2}\Big) \le \frac{4\eta_2^2\mathbb{E}(L_g^2)}{\varepsilon^2} \le \sigma_\Lambda^2\Big(\frac{2\eta_2}{\varepsilon}\Big)^2. \tag{3.33}$$

By union bound and Lemma 3.9.5, we get

$$\begin{aligned} \mathrm{P}\Big(\bigcup_{i=1}^{T_2}\{|g(x_i)| \ge \varepsilon/2\}\Big) &\le \sum_{i=1}^{T_2}\mathrm{P}\Big(|g(x_i)| \ge \varepsilon/2\Big) \\ &\le 2p\Big(\frac{2\sqrt{2}\ell}{\eta_2}\Big)^d \exp\Big(-\frac{\varepsilon^2\|v\|_2^2}{4+4\varepsilon\|v\|_\infty/3}\Big). \end{aligned} \tag{3.34}$$

If $|g(x_i)| < \varepsilon/2$ for all i and $L_g < \varepsilon/2\eta_2$, then $|g(x)| < \varepsilon$ for all $x \in \mathscr{X}^2$. It follows immediately

from (3.33) and (3.34) that

$$\begin{aligned}\mathrm{P}\Big(\sup_{x\in\mathscr{X}}\|V_{\Sigma\Delta}z(x)\|_1 < p\varepsilon\|v\|_2^2\Big) &= \mathrm{P}\Big(\sup_{x\in\mathscr{X}}|g(x)| < \varepsilon\Big) \\ &\geq 1-\sigma_\Lambda^2\Big(\frac{2\eta_2}{\varepsilon}\Big)^2 - 2p\Big(\frac{2\sqrt{2}\ell}{\eta_2}\Big)^d \exp\Big(-\frac{\varepsilon^2\|v\|_2^2}{4+4\varepsilon\|v\|_\infty/3}\Big).\end{aligned} \tag{3.35}$$

Because $\|V_{\Sigma\Delta}D^r\|_\infty = 2^r$ and $\|u_y\|_\infty \leq c(K,r) := c(K,1,r)$ in Proposition 3.6.1, we have

$$\begin{aligned}\big|\langle\widetilde{V}_{\Sigma\Delta}z(x),\widetilde{V}_{\Sigma\Delta}D^r u_y\rangle\big| &= \frac{2}{p\|v\|_2^2}\big|\langle V_{\Sigma\Delta}z(x),V_{\Sigma\Delta}D^r u_y\rangle\big| \\ &\leq \frac{2}{p\|v\|_2^2}\|V_{\Sigma\Delta}z(x)\|_1\|V_{\Sigma\Delta}D^r u_y\|_\infty \\ &\leq \frac{2}{p\|v\|_2^2}\|V_{\Sigma\Delta}z(x)\|_1\|V_{\Sigma\Delta}D^r\|_\infty\|u_y\|_\infty \\ &\leq 2^{r+1}c(K,r)g(x).\end{aligned}$$

Therefore, one can get

$$\begin{aligned}&\mathrm{P}\Big(\sup_{x,y\in\mathscr{X}}\big|\langle\widetilde{V}_{\Sigma\Delta}z(x),\widetilde{V}_{\Sigma\Delta}D^r u_y\rangle\big| < \varepsilon\Big) \\ &\geq 1-\sigma_\Lambda^2\Big(\frac{c(K,r)2^{r+2}\eta_2}{\varepsilon}\Big)^2 - 2p\Big(\frac{2\sqrt{2}\ell}{\eta_2}\Big)^d \exp\Big(-\frac{\varepsilon^2\|v\|_2^2}{c(K,r)^2 2^{2r+4}+c(K,r)2^{r+3}\varepsilon\|v\|_\infty/3}\Big).\end{aligned}$$

(ii) By repeating the statements before (3.35) with $V_{\Sigma\Delta}$ replaced with V_β, one can get

$$\mathrm{P}\Big(\sup_{x\in\mathscr{X}}\|V_\beta z(x)\|_1 < p\varepsilon\|v_\beta\|_2^2\Big) \geq 1-\sigma_\Lambda^2\Big(\frac{2\eta_2}{\varepsilon}\Big)^2 - 2p\Big(\frac{2\sqrt{2}\ell}{\eta_2}\Big)^d \exp\Big(-\frac{\varepsilon^2\|v_\beta\|_2^2}{4+4\varepsilon\|v_\beta\|_\infty/3}\Big). \tag{3.36}$$

Due to $\|V_\beta H\|_\infty = \beta^{-\lambda}$ and $\|u_y\|_\infty \le c(K,\beta)$ in Proposition 3.6.2, we get

$$\begin{aligned}\big|\langle \widetilde{V}_\beta z(x), \widetilde{V}_\beta H u_y\rangle\big| &= \frac{2}{p\|v_\beta\|_2^2}\big|\langle V_\beta z(x), V_\beta H u_y\rangle\big| \\ &\le \frac{2}{p\|v_\beta\|_2^2}\|V_\beta z(x)\|_1 \|V_\beta H u_y\|_\infty \\ &\le \frac{2}{p\|v_\beta\|_2^2}\|V_\beta z(x)\|_1 \|V_\beta H\|_\infty \|u_y\|_\infty \\ &\le \frac{2\beta^{-\lambda} c(K,\beta)}{p\|v_\beta\|_2^2}\|V_\beta z(x)\|_1.\end{aligned} \tag{3.37}$$

It follows from (3.36), (3.37) that

$$\begin{aligned}&\mathrm{P}\Big(\sup_{x,y\in\mathscr{X}}\big|\langle \widetilde{V}_\beta z(x), \widetilde{V}_\beta H u_y\rangle\big| < \varepsilon\Big) \\ &\ge 1 - \sigma_\Lambda^2\Big(\frac{4\eta_2 c(K,\beta)}{\varepsilon\beta^\lambda}\Big)^2 - 2p\Big(\frac{2\sqrt{2}\ell}{\eta_2}\Big)^d \exp\Big(-\frac{\varepsilon^2\|v_\beta\|_2^2\beta^{2\lambda}}{16c(K,\beta)^2 + 8\beta^\lambda c(K,\beta)\varepsilon\|v_\beta\|_\infty/3}\Big).\end{aligned}$$

□

3.9.4 Upper Bound of (IV)

Theorem 3.9.10. *Let $r \in \mathbb{N}^+$ and $\beta \in (1,2)$.*

1. *If u_x, u_y are state vectors of the $\Sigma\Delta$ quantizer $Q_{\Sigma\Delta}^{(r)}$, then*

$$\big|\langle \widetilde{V}_{\Sigma\Delta} D^r u_x, \widetilde{V}_{\Sigma\Delta} D^r u_y\rangle\big| \le \frac{c(K,r)^2 c(r)}{\lambda^{2r-1}},$$

   where $c(K,r)$ is the upper bound of the ℓ_∞ norm of state vectors in Proposition 3.6.1 and $c(r) > 0$ is a constant related to r.

2. *If u_x, u_y are state vectors of the noise-shaping quantizer Q_β, then*

$$\big|\langle \widetilde{V}_\beta H u_x, \widetilde{V}_\beta H u_y\rangle\big| \le \frac{2c(K,\beta)^2}{\beta^{2\lambda-2}},$$

where $c(K,\beta)$ is the upper bound of the ℓ_∞ norm of state vectors in Proposition 3.6.2.

Proof. (i) Cauchy-Schwarz inequality implies

$$\left|\langle \widetilde{V}_{\Sigma\Delta} D^r u_x, \widetilde{V}_{\Sigma\Delta} D^r u_y \rangle\right| \leq \frac{2}{p\|v\|_2^2} \|V_{\Sigma\Delta} D^r u_x\|_2 \|V_{\Sigma\Delta} D^r u_y\|_2.$$

One can easily verify that $V_{\Sigma\Delta} D^r$ is a sparse matrix such that each row has at most $r+1$ nonzero entries $\{w_0, w_1, \ldots, w_r\}$ of the following form

$$w_k = (-1)^{r+k} \binom{r}{k}.$$

Since $\max\{\|u_x\|_\infty, \|u_y\|_\infty\} \leq c(K,r)$ as indicated by Proposition 3.6.1, we have $\|V_{\Sigma\Delta} D^r u_x\|_2 \leq c(K,r)c(r)\sqrt{p}$ and $\|V_{\Sigma\Delta} D^r u_y\|_2 \leq c(K,r)c(r)\sqrt{p}$. So above inequality becomes

$$\left|\langle \widetilde{V}_{\Sigma\Delta} D^r u_x, \widetilde{V}_{\Sigma\Delta} D^r u_y \rangle\right| \leq \frac{2}{p\|v\|_2^2} \|V_{\Sigma\Delta} D^r u_x\|_2 \|V_{\Sigma\Delta} D^r u_y\|_2 \leq \frac{2c(r)^2 c(K,r)^2}{\|v\|_2^2} \leq \frac{c(K,r)^2 c'(r)}{\lambda^{2r-1}}$$

where the last inequality is due to $\|v\|_2^2 \geq \lambda^{2r-1} r^{-2r}$.

(ii) In the case of noise-shaping quantization, similarly, we have

$$\left|\langle \widetilde{V}_\beta H u_x, \widetilde{V}_\beta H u_y \rangle\right| \leq \frac{2}{p\|v_\beta\|_2^2} \|V_\beta H u_x\|_2 \|V_\beta H u_y\|_2.$$

Note that $V_\beta H = (I_p \otimes v_\beta)(I_p \otimes H_\beta) = I_p \otimes (v_\beta H_\beta)$ with $v_\beta H_\beta = (0,0,\ldots,0,\beta^{-\lambda}) \in \mathbb{R}^{1\times\lambda}$, and $\max\{\|u_x\|_\infty, \|u_y\|_\infty\} \leq c(K,\beta)$ by Proposition 3.6.2. It follows that $\|V_\beta H u_x\|_2 \leq \beta^{-\lambda}\sqrt{p}\|u_x\|_\infty$ and $\|V_\beta H u_y\|_2 \leq \beta^{-\lambda}\sqrt{p}\|u_y\|_\infty$. Then one can get

$$\left|\langle \widetilde{V}_\beta H u_x, \widetilde{V}_\beta H u_y \rangle\right| \leq \frac{2}{p\|v_\beta\|_2^2} \|V_\beta H u_x\|_2 \|V_\beta H u_y\|_2 \leq \frac{2\beta^{-2\lambda} c(K,\beta)^2}{\|v_\beta\|_2^2} \leq \frac{2c(K,\beta)^2}{\beta^{2\lambda-2}}$$

where the last inequality comes from $\|v_\beta\|_2 \geq \beta^{-1}$. □

3.9.5 Proof of Theorem 3.3.1

Proof. Recall that the kernel approximation errors in (3.23) and (3.24) can be bounded by four terms (I), (II), (III), (IV).

(i) For the $\Sigma\Delta$ scheme, in Theorem 3.9.8, we choose $\varepsilon = O(\sqrt{p^{-1}\log p})$, $\lambda = O(\sqrt{p\log^{-1} p})$ and $\eta_1 = O(p^{-2-\alpha})$ with $\alpha > 0$. Moreover, since $\|v\|_2^2 \geq \lambda^{2r-1} r^{-2r}$ and $\|v\|_\infty = O(\lambda^{r-1})$ (see Lemma 4.6 in [21]), in Theorem 3.9.9, we can choose $\varepsilon = O(c(K,r)\lambda^{-r+1}\log^{1/2} p)$ and $\eta_2 = O(\lambda^{-r-1}\log^{1/2} p)$. Then (3.16) follows immediately by combining above results with part (1) in Theorem 3.9.10.

(ii) As for the noise-shaping scheme, in Theorem 3.9.8, we choose the same parameters as in part (i): $\varepsilon = O(\sqrt{p^{-1}\log p})$, $\lambda = O(\sqrt{p\log^{-1} p})$ and $\eta_1 = O(p^{-2-\alpha})$ with $\alpha > 0$. Nevertheless, according to $\|v_\beta\|_2^2 \geq \beta^{-2}$ and $\|v_\beta\|_\infty = \beta^{-1}$, we set different values $\varepsilon = O(c(K,\beta)\beta^{-\lambda+1}\sqrt{p})$ and $\eta_2 = O(p^{-1})$ in Theorem 3.9.9. Therefore, (3.17) holds by applying above results and part (2) in Theorem 3.9.10. □

3.10 Proof of theorem 3.3.3

The architecture for the proof of theorem 3.3.3 closely follows the methods used in [40]. We start with some useful lemmata that aid in proving theorem 3.3.3.

Given a b-bit alphabet as in (3.4) with $b = \log_2(2K)$, we consider the following first-order $\Sigma\Delta$ quantization scheme for a random Fourier feature vector $z(x) \in [-1,1]^m$ corresponding to a data point $x \in \mathbb{R}^d$, where, the state variable $(u_x)_0$ is initialized as a random number, i.e.

$$(u_x)_0 \sim U\left[-\frac{1}{2^b-1}, \frac{1}{2^b-1}\right]$$

$$q_{k+1} = Q_{MSQ}((z(x))_{k+1} + (u_x)_k)$$

$$(u_x)_{k+1} = (u_x)_k + (z(x)))_{k+1} - q_{k+1}$$

The corresponding recurrence equation can written as

$$\tilde{V}Q(z(x)) = \tilde{V}z(x) - \tilde{V}Du_x + \tilde{V}(u_0^x, 0, \ldots, 0)^\top.$$

Lemma 3.10.1. *Given the following first order Sigma-Delta quantization scheme with a b-bit alphabet as in* (3.4)*, for a vector $z \in \mathbb{R}^m$ with $z \in [-1,1]^m$,*

$$\begin{aligned} u_0 &\sim U\left[-\frac{1}{2^b-1}, \frac{1}{2^b-1}\right] \\ q_{k+1} &= Q_{MSQ}(z_{k+1} + u_k) \\ u_{k+1} &= u_k + z_{k+1} - q_{k+1}, \end{aligned}$$

for each $k = 0, 1, \cdots m-1$, we have $u_k \sim U\left[-\frac{1}{2^b-1}, \frac{1}{2^b-1}\right]$.

Proof. Let the inductive hypothesis be $u_k \sim U\left[-\frac{1}{2^b-1}, \frac{1}{2^b-1}\right]$. Note that this is true by definition for u_0.

Case: $\frac{j}{2^b-1} \leq z_{k+1} \leq \frac{j+1}{2^b-1}$ where $j \in \{1, 3, \cdots 2^b-3\}$.

$u_k \sim U\left[-\frac{1}{2^b-1}, \frac{1}{2^b-1}\right]$ implies that $z_{k+1} + u_k \sim U\left[-\frac{1}{2^b-1} + z_{k+1}, \frac{1}{2^b-1} + z_{k+1}\right]$. Since by assumption, $\frac{j}{2^b-1} \leq z_{k+1} \leq \frac{j+1}{2^b-1}$ we see that $z_{k+1} + u_k \in [\frac{j-1}{2^b-1}, \frac{j+2}{2^b-1}]$ and thus

$$Q_{MSQ}(z_{k+1} + u_k) = \begin{cases} \frac{j}{2^b-1} & \text{if } \frac{j-1}{2^b-1} \leq z_{k+1} + u_k \leq \frac{j+1}{2^b-1}, \\ \frac{j+2}{2^b-1} & \text{if } \frac{j+1}{2^b-1} \leq z_{k+1} + u_k \leq \frac{j+2}{2^b-1}, \end{cases}$$

which in turn implies that

$$u_{k+1} = \begin{cases} z_{k+1} + u_k - \frac{j}{2^b-1} & \text{if } \frac{j-1}{2^b-1} \leq z_{k+1} + u_k \leq \frac{j+1}{2^b-1}, \\ z_{k+1} + u_k - \frac{j+2}{2^b-1} & \text{if } \frac{j+1}{2^b-1} \leq z_{k+1} + u_k \leq \frac{j+2}{2^b-1}. \end{cases}$$

Now we can compute the CDF of u_{k+1} (conditioned on z) as follows

$$\begin{aligned}
\mathbb{P}(u_{k+1} \le \alpha \,|\, z) &= \mathbb{P}\Big(z_{k+1} + u_k - \frac{j}{2^b-1} \le \alpha\,,\, q_k = \frac{j}{2^b-1} \,\Big|\, z\Big) \\
&\quad + \mathbb{P}\Big(z_{k+1} + u_k - \frac{j+2}{2^b-1} \le \alpha\,,\, q_k = \frac{j+2}{2^b-1} \,\Big|\, z\Big) \\
&= \mathbb{P}\Big(\frac{j-1}{2^b-1} - z_{k+1} \le u_k \le \min\Big\{\frac{j}{2^b-1} + \alpha - z_{k+1}, \frac{j+1}{2^b-1} - z_{k+1}\Big\} \,\Big|\, z\Big) \\
&\quad + \mathbb{P}\Big(\frac{j+1}{2^b-1} - z_{k+1} \le u_k \le \min\Big\{\frac{j+2}{2^b-1} + \alpha - z_{k+1}, \frac{j+2}{2^b-1} - z_{k+1}\Big\} \,\Big|\, z\Big) \\
&= \mathbb{P}\Big(u_k \le \min\Big\{\frac{j}{2^b-1} + \alpha - z_{k+1}, \frac{j+1}{2^b-1}\Big\} \,\Big|\, z\Big) \\
&\quad + \mathbb{P}\Big(\frac{j+1}{2^b-1} - z_{k+1} \le u_k \le \min\Big\{\frac{j+2}{2^b-1} + \alpha - z_{k+1}, \frac{j+2}{2^b-1} - z_{k+1}\Big\} \,\Big|\, z\Big).
\end{aligned}$$

Note that in the third equality we make use of the fact that $\frac{j}{2^b-1} \le z_{k+1} \le \frac{j+1}{2^b-1}$ implies $\frac{j-1}{2^b-1} - z_{k+1} \le -\frac{1}{2^b-1}$. Now note that

$$\begin{aligned}
&\mathbb{P}\Big(u_k \le \min\Big\{\frac{j}{2^b-1} + \alpha - z_{k+1}, \frac{j+1}{2^b-1}\Big\} \,\Big|\, z\Big) \\
&= \begin{cases}
0 & \text{if } \alpha < z_{k+1} - \frac{j+1}{2^b-1}, \\
\int_{-\frac{1}{2^b-1}}^{\frac{j}{2^b-1}+\alpha-z_{k+1}} \frac{2^b-1}{2} = \frac{2^b-1}{2}\left(\frac{j+1}{2^b-1} + \alpha - z_{k+1}\right) & \text{if } z_{k+1} - \frac{j+1}{2^b-1} \le \alpha < \frac{1}{2^b-1}, \\
\int_{-\frac{1}{2^b-1}}^{\frac{j+1}{2^b-1}-z_{k+1}} \frac{2^b-1}{2} = \frac{2^b-1}{2}\left(\frac{j+2}{2^b-1} - z_{k+1}\right) & \text{if } \alpha \ge \frac{1}{2^b-1},
\end{cases}
\end{aligned}$$

and

$$\begin{aligned}
&\mathbb{P}\Big(\frac{j+1}{2^b-1} - z_{k+1} \le u_k \le \min\Big\{\frac{j+2}{2^b-1} + \alpha - z_{k+1}, \frac{j+2}{2^b-1} - z_{k+1}\Big\} \,\Big|\, z\Big) \\
&= \begin{cases}
0 & \text{if } \alpha < -\frac{1}{2^b-1}, \\
\int_{\frac{j+1}{2^b-1}-z_{k+1}}^{\frac{j+2}{2^b-1}+\alpha-z_{k+1}} \frac{2^b-1}{2} = \frac{2^b-1}{2}\left(\frac{1}{2^b-1} + \alpha\right) & \text{if } -\frac{1}{2^b-1} \le \alpha < z_{k+1} - \frac{j+1}{2^b-1}, \\
\int_{\frac{j+1}{2^b-1}-z_{k+1}}^{\frac{1}{2^b-1}} \frac{2^b-1}{2} = \frac{2^b-1}{2}\left(z_{k+1} - \frac{j}{2^b-1}\right) & \text{if } \alpha \ge z_{k+1} - \frac{j+1}{2^b-1}.
\end{cases}
\end{aligned}$$

Thus

$$\mathbb{P}(u_{k+1} \leq \alpha \,|\, z) = \begin{cases} 0 & \text{if } \alpha < -\frac{1}{2^b-1}, \\ \frac{2^b-1}{2}\left(\frac{1}{2^b-1}+\alpha\right) & \text{if } -\frac{1}{2^b-1} \leq \alpha < z_{k+1} - \frac{j+1}{2^b-1}, \\ \frac{2^b-1}{2}\left(\frac{1}{2^b-1}+\alpha\right) & \text{if } z_{k+1} - \frac{j+1}{2^b-1} \leq \alpha < \frac{1}{2^b-1}, \\ 0 & \text{if } \alpha \geq \frac{1}{2^b-1}, \end{cases}$$

which shows that $u_{k+1} \,|\, z \sim U\left[-\frac{1}{2^b-1}, \frac{1}{2^b-1}\right]$.

Showing that $u_{k+1} \,|\, z \sim U\left[-\frac{1}{2^b-1}, \frac{1}{2^b-1}\right]$ for the other cases, namely, $\frac{j}{2^b-1} \leq z_{k+1} \leq \frac{j+1}{2^b-1}$ where $j \in \{0,2,\cdots 2^b-2\}$ and $-\frac{j+1}{2^b-1} \leq z_{k+1} \leq -\frac{j}{2^b-1}$ where $j \in \{1,3,\cdots 2^b-3\}$ and $-\frac{j+1}{2^b-1} \leq z_{k+1} \leq -\frac{j}{2^b-1}$ where $j \in \{0,2,\cdots 2^b-2\}$ follow a similar argument as above and for the sake of brevity is skipped from an explicit mention. Thus, by induction, we have shown that $u_{k+1} \,|\, z \sim U\left[-\frac{1}{2^b-1}, \frac{1}{2^b-1}\right]$. □

For the subsequent sections, we adopt the following notations . Let A be the matrix whose rows are the vectors $\{(\tilde{V}z(x))^T\}_x$, B be the matrix whose rows are the vectors $\{(\tilde{V}D^r u_x)^T\}_x$ and C be the matrix whose first column is $\frac{\sqrt{2}}{\sqrt{p}\|v\|_2}(u_0^{x_1}\, u_0^{x_2} \cdots u_0^{x_n})^T$ and all other columns as zero. Let the columns of A, B, C be denoted by A_i, B_i, C_i respectively. Now the corresponding approximation to the kernel can written as

$$\hat{K}_{\Sigma\Delta} = (A-B+C)(A-B+C)^T = \sum_{i=1}^{p}(A_i - B_i + C_i)(A_i - B_i + C_i)^T.$$

Lemma 3.10.2.

$$\mathbb{E}[\hat{K}_{\Sigma\Delta}] = K + \sum_{i=1}^{p}\Lambda_i \preccurlyeq K + \frac{1}{\lambda(2^b-1)^2}\left(8 + \frac{26}{3p}\right)I$$

where each Λ_i is a diagonal matrix with positive diagonal entries, $\delta > 0$ and I is the identity matrix.

Proof. We begin by noting that

$$\begin{aligned}
\mathbb{E}[\hat{K}_{\Sigma\Delta}] &= \mathbb{E}[\sum_{i=1}^{p}(A_i - B_i + C_i)(A_i - B_i + C_i)^T] \\
&= \mathbb{E}[\sum_{i=1}^{p} A_i A_i^T] - \mathbb{E}[\sum_{i=1}^{p} A_i B_i^T] - \mathbb{E}[\sum_{i=1}^{p} B_i A_i^T] + \mathbb{E}[\sum_{i=1}^{p} B_i B_i^T] + \mathbb{E}[\sum_{i=1}^{p} A_i C_i^T] \\
&+ \mathbb{E}[\sum_{i=1}^{p} C_i A_i^T] - \mathbb{E}[\sum_{i=1}^{p} B_i C_i^T] - \mathbb{E}[\sum_{i=1}^{p} C_i B_i^T] + \mathbb{E}[\sum_{i=1}^{p} C_i C_i^T] \\
&= K - \sum_{i=1}^{p} \mathbb{E}[A_i B_i^T] - \sum_{i=1}^{p} \mathbb{E}[B_i A_i^T] + \sum_{i=1}^{p} \mathbb{E}[B_i B_i^T] + \mathbb{E}[B_1 C_1^T] + \mathbb{E}[C_1 B_1^T] + \mathbb{E}[C_1 C_1^T]
\end{aligned}$$

where we've used the result from lemma 3.9.3 that $\mathbb{E}[AA^T] = K$. Now, let $F := \tilde{V} D^r \in \mathbb{R}^{p \times m}$ and $\{f_i^T\}_{i=1}^{p}$ denote the rows of F. Then let

$$B_i := \begin{pmatrix} f_i^T u_{x_1} \\ f_i^T u_{x_2} \\ \vdots \\ f_i^T u_{x_n} \end{pmatrix}$$

where B_i is the i-th column of B, $\{x_1, \cdots x_n\}$ represent the individual data samples and $u_{x_j} \in \mathbb{R}^m$ for each $j = 1, \cdots, n$. Note that u_{x_j} (when conditioned on Z) across data points x_j are independent with respect to each other with their entries $\sim U[-1/(2^b - 1), 1/(2^b - 1)]$. Thus,

$$\begin{aligned}
\mathbb{E}[A_i B_i^T] &= \mathbb{E}[A_i \begin{pmatrix} f_i^T u_{x_1} & f_i^T u_{x_2} & \cdots & f_i^T u_{x_n} \end{pmatrix}] \\
&= \mathbb{E}_Z[A_i \begin{pmatrix} \mathbb{E}_{u_{x_1}}[f_i^T u_{x_1} \,|\, z_{x_1}] & \mathbb{E}_{u_{x_2}}[f_i^T u_{x_2} \,|\, z_{x_2}] & \cdots & \mathbb{E}_{u_{x_n}}[f_i^T u_{x_n} \,|\, z_{x_n}] \end{pmatrix}] \\
&= 0.
\end{aligned}$$

By a similar argument, $\mathbb{E}[B_i A_i^T] = 0$ and $\mathbb{E}[\sum_{i=1}^p A_i C_i^T] = \mathbb{E}[\sum_{i=1}^p C_i A_i^T] = 0$. Now,

$$\sum_{i=1}^{p} \mathbb{E}[B_i B_i^T] = \sum_{i=1}^{p} \mathbb{E}_Z \mathbb{E}_U \left[\begin{pmatrix} (f_i^T u_{x_1})(f_i^T u_{x_1}) & (f_i^T u_{x_1})(f_i^T u_{x_2}) & \cdots & (f_i^T u_{x_1})(f_i^T u_{x_n}) \\ (f_i^T u_{x_2})(f_i^T u_{x_1}) & (f_i^T u_{x_2})(f_i^T u_{x_2}) & \cdots & (f_i^T u_{x_2})(f_i^T u_{x_n}) \\ \vdots & \vdots & \vdots & \vdots \\ (f_i^T u_{x_n})(f_i^T u_{x_1}) & \cdots & \cdots & (f_i^T u_{x_n})(f_i^T u_{x_n}) \end{pmatrix} \right].$$

First we note that u_{x_j} (when conditioned on Z) across data points x_j are independent with respect to each other with their entries $\sim U[-1/(2^b-1), 1/(2^b-1)]$ and thus $\mathbb{E}[BB^T]$ is a diagonal matrix. Then note that each row of VD has atmost 2 non-zero entries which are either $\{1,-1\}$. Thus,

$$|f_i^T u_{x_i}| = |\langle f_i, u_{x_j} \rangle| \leq \frac{\sqrt{2}}{\sqrt{p}\|v\|_2} \frac{2}{2^b-1} = \frac{2^{3/2}}{\sqrt{p}(2^b-1)\|v\|_2}.$$

Further, since $r=1$, $\|v\|_2^2 = \lambda$ which implies $|f_i^T u_{x_i}| \leq \frac{2^{3/2}}{\sqrt{p\lambda}(2^b-1)}$. Thus, the diagonal matrix $\mathbb{E}[B_i B_i^T] \preccurlyeq \frac{8}{p\lambda(2^b-1)^2} I$ in turn implies $\mathbb{E}[BB^T] \preccurlyeq \frac{8}{\lambda(2^b-1)^2} I$. Now, we have

$$\mathbb{E}[B_1 C_1^T] = \frac{\sqrt{2}}{\sqrt{p}\|v\|_2} \mathbb{E}_Z \mathbb{E}_U \left[\begin{pmatrix} (f_1^T u_{x_1})(u_0^{x_1}) & (f_1^T u_{x_1})(u_0^{x_2}) & \cdots & (f_1^T u_{x_1})(u_0^{x_n}) \\ (f_1^T u_{x_2})(u_0^{x_1}) & (f_1^T u_{x_2})(u_0^{x_2}) & \cdots & (f_1^T u_{x_2})(u_0^{x_n}) \\ \vdots & \vdots & \vdots & \vdots \\ (f_1^T u_{x_n})(u_0^{x_1}) & \cdots & \cdots & (f_1^T u_{x_n})(u_0^{x_n}) \end{pmatrix} \right].$$

Thus, by similar reasoning as in prior paragraphs, $\mathbb{E}[B_1 C_1^T]$ is a diagonal matrix and also

$$|u_0^x (f_1^T u_x)| \leq \frac{1}{2^b-1} |f_1^T u_x| \leq \frac{2^{3/2}}{\sqrt{p\lambda}(2^b-1)^2}$$

and thus

$$\frac{\sqrt{2}}{\sqrt{p}\|v\|_2} |u_0^x (f_1^T u_x)| \leq \frac{4}{p\lambda(2^b-1)^2}.$$

So $\mathbb{E}[-B_1C_1^T] \preccurlyeq \frac{4}{p\lambda(2^b-1)^2}I$. Similarly, $\mathbb{E}[-C_1B_1^T] \preccurlyeq \frac{4}{p\lambda(2^b-1)^2}I$. Now,

$$\begin{aligned}\mathbb{E}[C_1C_1^T] &= \frac{2}{p\|v\|_2^2}\mathbb{E}_Z\mathbb{E}_U\left[\begin{pmatrix}(u_0^{x_1})(u_0^{x_1}) & (u_0^{x_1})(u_0^{x_2}) & \cdots & (u_0^{x_1})(u_0^{x_n})\\ (u_0^{x_2})(u_0^{x_1}) & (u_0^{x_2})(u_0^{x_2}) & \cdots & (u_0^{x_2})(u_0^{x_n})\\ \vdots & \vdots & \vdots & \vdots \\ (u_0^{x_n})(u_0^{x_1}) & \cdots & \cdots & (u_0^{x_n})(u_0^{x_n})\end{pmatrix}\right]\\ &= \frac{2}{3p\lambda(2^b-1)^2}\end{aligned}$$

and thus $\mathbb{E}[C_1C_1^T] \preccurlyeq \left(\frac{2}{3r^{-2r}}\right)\frac{1}{p\lambda^{2r-1}}$. Thus, putting together the bounds for each of the terms, we get

$$\begin{aligned}\mathbb{E}[\hat{K}_{\Sigma\Delta}] &= K + \mathbb{E}[BB^T] - \mathbb{E}[B_1C_1^T] - \mathbb{E}[C_1B_1^T] + \mathbb{E}[C_1C_1^T]\\ &= K + \Lambda\end{aligned}$$

where $\Lambda := \mathbb{E}[BB^T] - \mathbb{E}[B_1C_1^T] - \mathbb{E}[C_1B_1^T] + \mathbb{E}[C_1C_1^T]$ is a diagonal matrix and

$$\Lambda \preccurlyeq \frac{1}{\lambda(2^b-1)^2}\left[8 + \frac{8}{p} + \frac{2}{3p}\right]I.$$

Thus $\mathbb{E}[\Lambda] \preccurlyeq \delta I$ where

$$\delta := \frac{1}{\lambda(2^b-1)^2}\left[8 + \frac{26}{3p}\right].$$

□

Lemma 3.10.3 ([40])**.** *Let $\eta > 0$, K and $\hat{K}$ be positive symmetric semi-definite matrices, then*

$$(1-\Delta_1)(K+\eta I) \preccurlyeq (\hat{K}+\eta I) \preccurlyeq (1+\Delta_2)(K+\eta I) \iff -\Delta_1 I \preccurlyeq M(\hat{K}-K)M \preccurlyeq \Delta_2 I$$

where, $M := (K+\eta I)^{-1/2}$.

Proof. The proof is obtained using the following sequence of equivalent statements.

$$
\begin{aligned}
& (1-\Delta_1)(K+\eta I) \preccurlyeq (\hat{K}+\eta I) \preccurlyeq (1+\Delta_2)(K+\eta I) \\
\iff & (1-\Delta_1) I \preccurlyeq (K+\eta I)^{-1/2}(\hat{K}+\eta I)(K+\eta I)^{-1/2} \preccurlyeq (1+\Delta_2) I \\
\iff & -\Delta_1 I \preccurlyeq M(\hat{K}+\eta I)M - I \preccurlyeq \Delta_2 I \\
\iff & -\Delta_1 I \preccurlyeq M(\hat{K}+\eta I)M - (K+\eta I)^{-1/2}(K+\eta I)(K+\eta I)^{-1/2} \preccurlyeq \Delta_2 I \\
\iff & -\Delta_1 I_n \preccurlyeq M(\hat{K}+\eta I - K - \eta I_n)M \preccurlyeq \Delta_2 I_n \\
\iff & -\Delta_1 I \preccurlyeq M(\hat{K}-K)M \preccurlyeq \Delta_2 I.
\end{aligned}
$$

Note that the assumptions made on $K, \hat{K}$ and η imply that $\hat{K}+\eta I$ is invertible and also $(K+\eta I)^{-1/2}$ exists. □

Lemma 3.10.4 ([40])**.** *Let $0 \preccurlyeq \Lambda \preccurlyeq \delta I$ where $\delta > 0$. Also let $\eta > 0$, K and $\hat{K}$ be positive symmetric semi-definite matrices and $M := (K+\eta I)^{-1/2}$. Then*

$$
-\Delta_1 I_n \preccurlyeq M(\hat{K}-(K+\Lambda))M \preccurlyeq (\Delta_2 - \frac{\delta}{\eta}) I_n \implies -\Delta_1 I_n \preccurlyeq M(\hat{K}-K)M \preccurlyeq \Delta_2 I_n.
$$

Proof. We begin by noting that $0 \preccurlyeq M\Lambda M$ since M is invertible, $0 \preccurlyeq \Lambda$ and for all $x \neq 0$, $x^T M\Lambda M x = (Mx)^T \Lambda (Mx)$. Thus

$$
-\Delta_1 I_n \preccurlyeq M(\hat{K}-(K+\Lambda))M \implies -\Delta_1 I_n \preccurlyeq M(\hat{K}-K)M.
$$

Additionally, note that $\|M\|_2^2 = \|M^2\|_2 = \|(K+\eta I)^{-1}\|_2$ where M and M^2 are symmetric, and $\|M\Lambda M\| \leq \|\Lambda\|\|(K+\eta I)^{-1}\|$. Also since $0 \preccurlyeq K$ (positive semi-definite kernel), we have that $\|(K+\eta I)^{-1}\|_2 \leq \frac{1}{\eta}$. Hence, we get

$$
-\Delta_1 I_n \preccurlyeq M(\hat{K}-(K+\Lambda))M \preccurlyeq (\Delta_2 - \frac{\delta}{\eta}) I_n \implies -\Delta_1 I_n \preccurlyeq M(\hat{K}-K)M \preccurlyeq \Delta_2 I_n.
$$

□

Theorem 3.10.5 (Matrix-Bernstein inequality [40]). *Consider a finite sequence $\{S_i\}$ of random Hermitian matrices of the same size and assume that*

$$\mathbb{E}[S_i] = 0 \quad \text{and} \quad \lambda_{max}(S_i) \leq l \quad \text{for each index } i.$$

Let $S = \sum_i S_i$ and $\mathbb{E}[S^2] \preccurlyeq W$, i.e. W is a semi-definite upper bound for the second moment of S. Then, for $t \geq 0$,

$$\mathbb{P}[\lambda_{max}(S) \geq t] \leq 4\frac{tr(W)}{\|W\|}.\exp(\frac{-t^2/2}{\|W\| + lt/3}).$$

Recall our notations where A is the matrix whose rows are the vectors $\{(\tilde{V}z(x))^T\}_x$, B is the matrix whose rows are the vectors $\{(\tilde{V}D^r u_x)^T\}_x$ and C is the matrix whose first column is $\frac{\sqrt{2}}{\sqrt{p}\|v\|_2}(u_0^{x_1}, u_0^{x_2}, \cdots, u_0^{x_n})^T$ and all other columns as zero. Also the columns of A, B, C are denoted by A_i, B_i, C_i respectively. Additionally let $K_i := \mathbb{E}[A_i A_i^T]$, $M := (K + \eta I_n)^{-1/2}$ and

$$S_i := M(A_i - B_i - C_i)(A_i - B_i - C_i)^T M^T - M(K_i + \Lambda_i)M^T. \tag{3.38}$$

Thus note that by design $\mathbb{E}[S_i] = 0$. We now will show that the remaining assumptions required to apply Matrix-Bernstein inequality hold for the sequence of matrices $\{S_i\}_{i=1}^p$.

Lemma 3.10.6. *The 2-norm of S_i (defined in* (3.38)*) is bounded for each $i = 1, \cdots, p$ and $\mathbb{E}[S]^2$ has a semi-definite upper bound, where $S = \sum_i S_i$. In particular,*

$$\|S_i\| \leq \frac{2n\lambda}{p\eta^2} \quad (:= l) \qquad \text{and} \qquad \mathbb{E}[S^2] \preccurlyeq l\tilde{M}$$

where, n is the number of data samples, η is the regularization, $m = \lambda p$ denotes the parameters of $\Sigma\Delta$ quantization and $\tilde{M} := M(K + \Lambda)M^T$.

Proof. (i) $\lambda_{max}(S_i)$ is bounded. Let $u_i := M(A_i - B_i - C_i)$, then $S_i = u_i u_i^T - \mathbb{E}[u_i u_i^T]$. First note

that

$$\begin{aligned}\|u_i u_i^T\| &= \|u_i\|^2 \\ &= \|M(A_i - B_i - C_i)\|^2 \\ &\leq \|M\|^2 \|A_i - B_i - C_i\|^2.\end{aligned}$$

Also, $A_i - B_i - C_i$ is the i-th column of the matrix which has as its rows the vectors $\{\tilde{V}Q(z(x))^T\}_x$. Thus, in general

$$A_i - B_i - C_i = \begin{pmatrix} g_i^T q_{x_1} \\ g_i^T q_{x_2} \\ \vdots \\ g_i^T q_{x_n} \end{pmatrix}$$

where, g_i^T denotes the i-th row of $\tilde{V}$. Also note that the entries of $Q(z(x))$ are in $\mathscr{A} = \left\{ \frac{a}{2K-1} \,\middle|\, a = \pm 1, \pm 3, \ldots, \pm(2K-1) \right\}$. Thus,

$$\|A_i - B_i - C_i\|_2^2 \leq n\|g_i\|_1^2 \leq n\|\tilde{V}\|_\infty^2.$$

Note that $\tilde{V} = \frac{\sqrt{2}}{\sqrt{p}\|v\|_2}(I_p \otimes v)$ where for $r = 1$, $v \in \mathbb{R}^\lambda$ is the vector of all ones, which implies $\|\tilde{V}\|_\infty = \frac{\sqrt{2\lambda}}{\sqrt{p}}$. Thus,

$$\|M\|^2 \|A_i - B_i - C_i\|^2 \leq \frac{2n\lambda}{p}\|M\|^2.$$

Further, since by definition $M = (K + \eta I)^{-1/2}$ we have,

$$\begin{aligned}\|M\|^2 \|A_i - B_i - C_i\|^2 &\leq \frac{2n\lambda}{p}\|M\|^2 \\ &= \frac{2n\lambda}{p}\|(K + \eta I)^{-1}\|^2 \\ &\leq \frac{2n\lambda}{p\eta^2} \quad (:= l).\end{aligned}$$

Thus we see that,

$$\begin{aligned}\|S_i\| &= \|u_i u_i^T - \mathbb{E}[u_i u_i^T]\| \\ &\leq \|u_i u_i^T\| + \|\mathbb{E}[u_i u_i^T]\| \\ &\leq 2l.\end{aligned}$$

So $\|S_i\| \leq 2l$ implies $\lambda_{max}(S_i) \leq 2l$. Note that K_i and Λ_i are expectations of symmetric matrices and thus S_i is symmetric.

(ii) $\mathbb{E}[S^2]$ has a semi-definite upper bound.

$$\begin{aligned}\mathbb{E}[S_i^2] &= \mathbb{E}[(u_i u_i^T)^2] - \mathbb{E}[u_i u_i^T]^2 \\ &\preccurlyeq \mathbb{E}[(u_i u_i^T)^2] = \mathbb{E}[\|u_i\|^2 u_i u_i^T] \\ &\preccurlyeq l\,\mathbb{E}[u_i u_i^T].\end{aligned}$$

Now,

$$\begin{aligned}\mathbb{E}[S^2] &= \sum_{i=1}^{p} \mathbb{E}[S_i^2] \\ &\preccurlyeq l \sum_{i=1}^{p} \mathbb{E}[u_i u_i^T] = l \sum_{i=1}^{p} M(K_i + \Lambda_i) M^T \\ &\preccurlyeq lM(K+\Lambda)M^T \\ &\preccurlyeq l\tilde{M}\end{aligned}$$

where $\tilde{M} := M(K+\Lambda)M^T$ and thus $\mathbb{E}[S^2] \preccurlyeq l\tilde{M}$. □

Now we are in a position to prove theorem 3.3.3 of the main text which we restate for convenience.

Theorem 3.10.7. *Let $\hat{K}_{\Sigma\Delta}$ be an approximation of a true kernel matrix K using m-feature first-order $\Sigma\Delta$ quantized RFF (as in* (3.18)*) with a b-bit alphabet (as in* (3.4)*) and $m = \lambda p$. Then*

given $\Delta_1 \geq 0, \Delta_2 \geq \frac{\delta}{\eta}$ *where* $\eta > 0$ *represents the regularization and* $\delta = \frac{8+\frac{26}{3p}}{\lambda(2^b-1)^2}$*, we have*

$$\begin{aligned}
&\mathbb{P}[(1-\Delta_1)(K+\eta I) \preccurlyeq (\hat{K}_{\Sigma\Delta}+\eta I) \preccurlyeq (1+\Delta_2)(K+\eta I)] \\
&\geq \quad 1-4n\left[\exp\left(\frac{-\Delta_1^2/2}{l(\frac{1}{\eta}(\|K\|_2+\delta)+2\Delta_1/3)}\right)+\exp\left(\frac{-(\Delta_2-\frac{\delta}{\eta})^2/2}{l(\frac{1}{\eta}(\|K\|_2+\delta)+2(\Delta_2-\frac{\delta}{\eta})/3)}\right)\right]
\end{aligned}$$

where, $l = \frac{2n\lambda}{p\eta^2}$.

Proof. We apply Matrix-Bernstein inequality (theorem 3.10.5) to $\{S_i\}_{i=1}^p$ (defined in 3.38) to obtain that given $t_2 \geq 0$,

$$\mathbb{P}[\lambda_{max}(M(\hat{K}_{\Sigma\Delta}-(K+\Lambda))M^T) \geq t_2] \leq 4\frac{\mathrm{tr}(\tilde{M})}{\|\tilde{M}\|}\exp\left(\frac{-t_2^2/2}{l\|\tilde{M}\|+2lt_2/3}\right).$$

Now, since $\lambda_{max}(S) = -\lambda_{min}(-S)$, by repeating an identical argument for $-S$ we obtain that given $t_1 \geq 0$,

$$\mathbb{P}[\lambda_{min}(M(\hat{K}_{\Sigma\Delta}-(K+\Lambda))M^T) \leq -t_1] \leq 4\frac{\mathrm{tr}(\tilde{M})}{\|\tilde{M}\|}\exp\left(\frac{-t_1^2/2}{l\|\tilde{M}\|+2lt_1/3}\right).$$

Putting the above two equations together with the fact that $M = (K+\eta I_n)^{-1/2}$ we obtain that for $t_1, t_2 \geq 0$,

$$\begin{aligned}
&\mathbb{P}[-t_1 I_n \preccurlyeq M(\hat{K}_{\Sigma\Delta}-(K+\Lambda))M \preccurlyeq t_2 I_n] \\
&\geq 1-4\frac{\mathrm{tr}(\tilde{M})}{\|\tilde{M}\|}\left[\exp\left(\frac{-t_1^2/2}{l\|\tilde{M}\|+2lt_1/3}\right)+\exp\left(\frac{-t_2^2/2}{l\|\tilde{M}\|+2lt_2/3}\right)\right].
\end{aligned}$$

Thus, by lemmas 3.10.2, 3.10.3, 3.10.4 and 3.10.6, for the $\Sigma\Delta$-quantized RFF kernel $\hat{K}_{\Sigma\Delta}$, given

$\Delta_1 \geq 0, \Delta_2 \geq \frac{\delta}{\eta}$ we have the following spectral approximation result:

$$\begin{aligned} & \mathbb{P}[(1-\Delta_1)(K+\eta I) \preccurlyeq (\hat{K}_{\Sigma\Delta}+\eta I) \preccurlyeq (1+\Delta_2)(K+\eta I)] \\ \geq \quad & 1-4\frac{\mathrm{tr}(\tilde{M})}{\|\tilde{M}\|}\left[\exp\left(\frac{-\Delta_1^2/2}{l(\|\tilde{M}\|+2\Delta_1/3)}\right)+\exp\left(\frac{-(\Delta_2-\frac{\delta}{\eta})^2/2}{l(\|\tilde{M}\|+2(\Delta_2-\frac{\delta}{\eta})/3)}\right)\right] \end{aligned}$$

where $\tilde{M} = M(K+\Lambda)M$, $M = (K+\eta I_n)^{-1/2}$, $l = \frac{2n\lambda}{p\eta^2}$ is the upper bound for $\|u_i u_i^T\|$ computed in lemma 3.10.6 and $\delta = \frac{8+\frac{26}{3p}}{\lambda(2^b-1)^2}$ is the bound computed in lemma 3.10.2 such that $\mathbb{E}[\hat{K}_{\Sigma\Delta}] \preccurlyeq K+\delta I$. Now, note that

$$\begin{aligned} \|\tilde{M}\|_2 &= \|M(K+\Lambda)M\|_2 \\ &\leq \|M\|_2^2\|K+\Lambda\|_2 \\ &\leq \|M\|_2^2(\|K\|_2+\|\Lambda\|_2) \\ &= \|(K+\eta I)^{-1}\|_2(\|K\|_2+\|\Lambda\|_2) \\ &\leq \frac{1}{\eta}(\|K\|_2+\delta). \end{aligned}$$

Also given the positive semi-definite matrix $\tilde{M}$, we know that $\|\tilde{M}\|_2 = \lambda_{max}(\tilde{M})$ and thus $tr(\tilde{M}) \leq \mathrm{rank}(\tilde{M})\|M\|_2$ which implies $\frac{\mathrm{tr}(\tilde{M})}{\|\tilde{M}\|} \leq \mathrm{rank}(\tilde{M}) \leq n$. Hence,

$$\begin{aligned} & \mathbb{P}[(1-\Delta_1)(K+\eta I) \preccurlyeq (\hat{K}_{\Sigma\Delta}+\eta I) \preccurlyeq (1+\Delta_2)(K+\eta I)] \\ \geq \quad & 1-4n\left[\exp\left(\frac{-\Delta_1^2/2}{l(\frac{1}{\eta}(\|K\|_2+\delta)+2\Delta_1/3)}\right)+\exp\left(\frac{-(\Delta_2-\frac{\delta}{\eta})^2/2}{l(\frac{1}{\eta}(\|K\|_2+\delta)+2(\Delta_2-\frac{\delta}{\eta})/3)}\right)\right]. \end{aligned}$$

□

3.11 Acknowledgements

Jinjie Zhang was partially supported by grants NSF DMS 2012546 and 2012266. Alexander Cloninger was partially supported by NSF DMS 1819222, 2012266. Rayan Saab was

partially supported by NSF DMS 2012546 and a UCSD senate research award. This chapter, in full, is joint work with Harish Kannan, Alexander Cloninger, Rayan Saab, and has been submitted for publication. The dissertation author was the primary investigator and author of this paper.

References

[1] Raj Agrawal, Trevor Campbell, Jonathan Huggins, and Tamara Broderick. "Data dependent compression of random features for large-scale kernel approximation". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 1822–1831.

[2] Ethem Alpaydin and Cenk Kaynak. "Cascading classifiers". In: *Kybernetika* 34.4 (1998), pp. 369–374.

[3] Haim Avron, Kenneth L Clarkson, and David P Woodruff. "Faster kernel ridge regression using sketching and preconditioning". In: *SIAM Journal on Matrix Analysis and Applications* 38.4 (2017), pp. 1116–1138.

[4] Haim Avron, Michael Kapralov, Cameron Musco, Christopher Musco, Ameya Velingker, and Amir Zandieh. "Random Fourier features for kernel ridge regression: Approximation bounds and statistical guarantees". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 253–262.

[5] Francis Bach. "On the equivalence between kernel quadrature rules and random feature expansions". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 714–751.

[6] John J Benedetto, Alexander M Powell, and Ozgur Yilmaz. "Sigma-delta quantization and finite frames". In: *IEEE Transactions on Information Theory* 52.5 (2006), pp. 1990–2005.

[7] John J Benedetto, Alexander M Powell, and Özgür Yılmaz. “Second-order sigma–delta ($\Sigma\Delta$) quantization of finite frame expansions”. In: *Applied and Computational Harmonic Analysis* 20.1 (2006), pp. 126–148.

[8] Petros T Boufounos and Shantanu Rane. “Efficient Coding of Signal Distances Using Universal Quantized Embeddings.” In: *DCC*. 2013, pp. 251–260.

[9] Evan Chou and C Sinan Güntürk. “Distributed noise-shaping quantization: I. Beta duals of finite frames and near-optimal quantization of random measurements”. In: *Constructive Approximation* 44.1 (2016), pp. 1–22.

[10] Evan Chou and C Sinan Güntürk. “Distributed noise-shaping quantization: II. Classical frames”. In: *Excursions in Harmonic Analysis, Volume 5*. Springer, 2017, pp. 179–198.

[11] Evan Chou, C Sinan Güntürk, Felix Krahmer, Rayan Saab, and Özgür Yılmaz. “Noise-shaping quantization methods for frame-based and compressive sampling systems”. In: *Sampling theory, a renaissance* (2015), pp. 157–184.

[12] Felipe Cucker and Steve Smale. “On the mathematical foundations of learning”. In: *Bulletin of the American mathematical society* 39.1 (2002), pp. 1–49.

[13] Ludwig Danzer. “” Helly’s theorem and its relatives,” in Convexity”. In: *Proc. Symp. Pure Math.* Vol. 7. Amer. Math. Soc. 1963, pp. 101–180.

[14] Ingrid Daubechies and Ron DeVore. “Approximating a bandlimited function using very coarsely quantized data: A family of stable sigma-delta modulators of arbitrary order”. In: *Annals of mathematics* (2003), pp. 679–710.

[15] Percy Deift, Felix Krahmer, and C Sınan Güntürk. “An optimal family of exponentially accurate one-bit Sigma-Delta quantization schemes”. In: *Communications on Pure and Applied Mathematics* 64.7 (2011), pp. 883–919.

[16] Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. Springer, 2013.

[17] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. “A kernel two-sample test”. In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 723–773.

[18] C Sinan Güntürk. “One-bit sigma-delta quantization with exponential accuracy”. In: *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 56.11 (2003), pp. 1608–1630.

[19] C Sinan Güntürk, Mark Lammers, Alexander M Powell, Rayan Saab, and Ö Yılmaz. “Sobolev duals for random frames and ΣΔ quantization of compressed sensing measurements”. In: *Foundations of Computational mathematics* 13.1 (2013), pp. 1–36.

[20] Thang Huynh. “Accurate quantization in redundant systems: From frames to compressive sampling and phase retrieval”. PhD thesis. New York University, 2016.

[21] Thang Huynh and Rayan Saab. “Fast binary embeddings and quantized compressed sensing with structured matrices”. In: *Communications on Pure and Applied Mathematics* 73.1 (2020), pp. 110–149.

[22] Felix Krahmer, Rayan Saab, and Rachel Ward. “Root-exponential accuracy for coarse quantization of finite frame expansions”. In: *IEEE transactions on information theory* 58.2 (2012), pp. 1069–1079.

[23] Xiaoyun Li and Ping Li. “Quantization algorithms for random fourier features”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 6369–6380.

[24] Chi-Jen Lin. *Large-scale kernel machines*. MIT press, 2007.

[25] Fanghui Liu, Xiaolin Huang, Yudong Chen, and Johan AK Suykens. “Random features for kernel approximation: A survey in algorithms, theory, and beyond”. In: *arXiv preprint arXiv:2004.11154* (2020).

[26] Lynn H Loomis. *Introduction to abstract harmonic analysis*. Courier Corporation, 2013.

[27] Avner May, Alireza Bagheri Garakani, Zhiyun Lu, Dong Guo, Kuan Liu, AurÃ©lien Bellet, Linxi Fan, Michael Collins, Daniel Hsu, Brian Kingsbury, Michael Picheny, and Fei Sha. “Kernel Approximation Methods for Speech Recognition”. In: *Journal of Machine Learning Research* 20.59 (2019), pp. 1–36. URL: http://jmlr.org/papers/v20/17-026.html.

[28] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[29] Ali Rahimi and Benjamin Recht. “Random features for large-scale kernel machines”. In: *Advances in neural information processing systems* 20 (2007), pp. 1177–1184.

[30] Alessandro Rudi and Lorenzo Rosasco. “Generalization Properties of Learning with Random Features.” In: *NIPS*. 2017, pp. 3215–3225.

[31] Vincent Schellekens and Laurent Jacques. “Breaking the waves: asymmetric random periodic features for low-bitrate kernel machines”. In: *arXiv preprint arXiv:2004.06560* (2020).

[32] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Adaptive Computation and Machine Learning series, 2018.

[33] John Shawe-Taylor, Nello Cristianini, et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[34] Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert RG Lanckriet. “Hilbert space embeddings and metrics on probability measures”. In: *The Journal of Machine Learning Research* 11 (2010), pp. 1517–1561.

[35] Bharath K Sriperumbudur and Zoltán Szabó. “Optimal rates for Random Fourier features”. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*. 2015, pp. 1144–1152.

[36] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[37] Dougal J Sutherland and Jeff Schneider. “On the error of random fourier features”. In: *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*. 2015, pp. 862–871.

[38] Stephen Tu, Rebecca Roelofs, Shivaram Venkataraman, and Benjamin Recht. “Large scale kernel learning using block coordinate descent”. In: *arXiv preprint arXiv:1602.05310* (2016).

[39] Lei Xu, Adam Krzyzak, and Ching Y Suen. “Methods of combining multiple classifiers and their applications to handwriting recognition”. In: *IEEE transactions on systems, man, and cybernetics* 22.3 (1992), pp. 418–435.

[40] Jian Zhang, Avner May, Tri Dao, and Christopher Ré. “Low-precision random Fourier features for memory-constrained kernel approximation”. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 1264–1274.

[41] Jinjie Zhang and Rayan Saab. “Faster Binary Embeddings for Preserving Euclidean Distances”. In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=YCXrx6rRCXO.

Chapter 4

Post-training Quantization for Neural Networks with Provable Guarantees

While neural networks have been remarkably successful in a wide array of applications, implementing them in resource-constrained hardware remains an area of intense research. By replacing the weights of a neural network with quantized (e.g., 4-bit, or binary) counterparts, massive savings in computation cost, memory, and power consumption are attained. To that end, we generalize a post-training neural-network quantization method, GPFQ, that is based on a greedy path-following mechanism. Among other things, we propose modifications to promote sparsity of the weights, and rigorously analyze the associated error. Additionally, our error analysis expands the results of previous work on GPFQ to handle general quantization alphabets, showing that for quantizing a single-layer network, the relative square error essentially decays linearly in the number of weights – i.e., level of over-parametrization. Our result holds across a range of input distributions and for both fully-connected and convolutional architectures thereby also extending previous results. To empirically evaluate the method, we quantize several common architectures with few bits per weight, and test them on ImageNet, showing only minor loss of accuracy compared to unquantized models. We also demonstrate that standard modifications, such as bias correction and mixed precision quantization, further improve accuracy.

4.1 Introduction

Over the past decade, deep neural networks (DNNs) have achieved great success in many challenging tasks, such as computer vision, natural language processing, and autonomous vehicles. Nevertheless, over-parameterized DNNs are computationally expensive to train, memory intensive to store, and energy consuming to apply. This hinders the deployment of DNNs to resource-limited applications. Therefore, model compression without significant performance degradation is an important active area of deep learning research [11, 6, 10]. One prominent approach to compression is *quantization*. Here, rather than adopt a 32-bit floating point format for the model parameters, one uses significantly fewer bits for representing weights, activations, and even gradients. Since the floating-point operations are substituted by more efficient low-bit operations, quantization can reduce inference time and power consumption.

Following [16], we can classify quantization methods into two categories: *quantization-aware training* and *post-training quantization*. The fundamental difficulty in quantization-aware training stems from the fact that it reduces to an integer programming problem with a non-convex loss function, making it NP-hard in general. Nevertheless, many well-performing heuristic methods exist, e.g., [4, 12, 35, 15, 33, 21, 31]. Here one, for example, either modifies the training procedure to produce quantized weights, or successively quantizes each layer and then retrains the subsequent layers. Retraining is a powerful, albeit computationally intensive way to compensate for the accuracy loss resulting from quantization and it remains generally difficult to analyze rigorously.

Hence, much attention has recently been dedicated to post-training quantization schemes, which directly quantize pretrained DNNs having real-valued weights, without retraining. These quantization methods either rely on a small amount of data [1, 3, 34, 23, 14, 30, 19, 22] or can be implemented without accessing training data, i.e. data-free compression [24, 2, 32, 20].

4.1.1 Related Work

We now summarize some prior work on post-training quantization methods. The majority of these methods aim to reduce quantization error by minimizing a mean squared error (MSE) objective, e.g. $\min_{\alpha>0}\left\|W-\alpha\left\lfloor\frac{W}{\alpha}\right\rceil\right\|_F$, where W is a weight matrix and $\lfloor\cdot\rceil$ is a round-off operator that represents a map from the set of real numbers to the low-bit alphabet. Generally $\lfloor\cdot\rceil$ simply assigns numbers in different intervals or "bins" to different elements of the alphabet. Algorithms in the literature differ in their choice of $\lfloor\cdot\rceil$, as they use different strategies for determining the quantization bins. However, they share the property that once the quantization bins are selected, weights are quantized independently of each other. For example, Banner, Nahshan, and Soudry [1] (see also [34]) choose the thresholds to minimize a MSE metric. Their numerical results also show that for convolutional networks using different quantization thresholds "per-channel" and bias correction can improve the accuracy of quantized models. Choukroun et al. [3] solve a minimum mean squared error (MMSE) problem for both weights and activations quantization. Based on a small calibration data set, Hubara et al. [14] suggest a per-layer optimization method followed by integer programming to determine the bit-width of different layers. A bit-split and stitching technique is used by [30] that "splits" integers into multiple bits, then optimizes each bit, and finally stitches all bits back to integers. Li et al. [19] leverage the basic building blocks in DNNs and reconstructs them one-by-one. As for data-free model quantization, there are different strategies, such as weight equalization [24], reconstructing calibration data samples according to batch normalization statistics (BNS) [2, 32], and adversarial learning [20].

4.1.2 Contribution

In spite of reasonable heuristic explanations and empirical results, all quantization methods mentioned in Section 4.1.1 lack rigorous theoretical guarantees. Recently, Lybrand and Saab [22] proposed and analyzed a method for quantizing the weights of pretrained DNNs called *greedy path following quantization* (GPFQ), see Section 4.2.2 for details. In this paper, we sub-

stantially improve GPFQ's theoretical analysis, propose a modification to handle convolutional layers, and propose a sparsity promoting version to encourage the algorithm to set many of the weights to zero. We demonstrate that the performance of our quantization methods is not only good in experimental settings, but, equally importantly, has favorable and rigorous error guarantees. Specifically, the contributions of this paper are threefold:

1. We generalize the results of [22] in several directions. Indeed, the results of [22] apply only to alphabets, $\mathscr{A}$, of the form $\mathscr{A} = \{0, \pm 1\}$ and standard Gaussian input because the proof technique in [22] relies heavily on properties of Gaussians and case-work over elements of the alphabet. It also requires the assumption that floating point weights are ε-away from alphabet elements. In contrast, by using a different and more natural proof technique, our results avoid this assumption and extend to general alphabets like $\mathscr{A}$ in (4.4) and make the main result in [22] a special case of our Theorem 4.3.4, which in turn follows from Theorem 4.3.1. Moreover, we extend the class of input vectors for which the theory applies. For example, in Section 4.3, we show that if the input data $X \in \mathbb{R}^{m \times N_0}$ is either bounded or drawn from a mixture of Gaussians, then the relative square error of quantizing a neuron $w \in \mathbb{R}^{N_0}$ satisfies the following inequality with high probability:

$$\frac{\|Xw - Xq\|_2^2}{\|Xw\|_2^2} \lesssim \frac{m \log N_0}{N_0} \tag{4.1}$$

where $q \in \mathscr{A}^{N_0}$ is the quantized weights. A mixture of Gaussians is a reasonable model for the output of some of the deeper layers in neural networks that focus on classification, thus our results are relevant in those contexts. Further, to handle convolutional neural networks (CNNs), we introduce a modification to GPFQ in Section 4.5.1 that relies on random subsampling to make quantizing DNNs practically feasible with large batch size m. This also allows us to obtain quantization error bounds that resemble (4.1), for single-layer CNNs in Section 4.3.3.

2. In order to reduce the storage, computational, and power requirements of DNNs one complimentary approach to quantization is to sparsify the weights, i.e., set many of them to zero. In Section 4.4, we propose modifications to GPFQ that leverage soft and hard thresholding

to increase sparsity of the weights of the *quantized* neural networks. We present error bounds, similar to the ones in Theorem 4.3.1, and provide their proofs in Section 4.10.

3. We provide extensive numerical experiments to illustrate the performance of GPFQ and its proposed modifications on common computer vision DNNs. First, we provide comparisons with other post-training quantization approaches (Section 4.5) and show that GPFQ achieves near-original model performance using 4 bits and that the results for 5 bits are competitive with state-of-the-art methods. Our experiments also demonstrate that GPFQ is compatible with various ad-hoc performance enhancing modifications such as bias correction [1], unquantizing the last layer [35, 19], and mixed precision [7, 2]. To illustrate the effects of sparsity, we further explore the interactions among prediction accuracy, sparsity of the weights, and regularization strength in our numerical experiments. Our results show that one can achieve near-original model performance even when half the weights (or more) are quantized to zero.

4.2 Preliminaries

In this section, we first introduce the notation that will be used throughout this paper and then recall the original GPFQ algorithm in [22].

4.2.1 Notation

Various positive absolute constants are denoted by C, c. We use $a \lesssim b$ as shorthand for $a \leq Cb$, and $a \gtrsim b$ for $a \geq Cb$. Let $S \subseteq \mathbb{R}^n$ be a Borel set. $\mathrm{Unif}(S)$ denotes the uniform distribution over S. An L-layer multi-layer perceptron, Φ, acts on a vector $x \in \mathbb{R}^{N_0}$ via

$$\Phi(x) := \varphi^{(L)} \circ A^{(L)} \circ \cdots \circ \varphi^{(1)} \circ A^{(1)}(x) \tag{4.2}$$

where $\varphi^{(i)} : \mathbb{R}^{N_i} \to \mathbb{R}^{N_i}$ is an activation function acting entrywise, and $A^{(i)} : \mathbb{R}^{N_{i-1}} \to \mathbb{R}^{N_i}$ is an affine map given by $A^{(i)}(z) := W^{(i)\top} z + b^{(i)}$. Here, $W^{(i)} \in \mathbb{R}^{N_{i-1} \times N_i}$ is a weight matrix and $b^{(i)} \in \mathbb{R}^{N_i}$ is a bias vector. Since $w^\top x + b = \langle (w,b), (x,1) \rangle$, the bias term $b^{(i)}$ can be treated as an

extra row to the weight matrix $W^{(i)}$, so we will henceforth ignore it. For theoretical analysis, we focus on infinite *mid-tread* alphabets with step size $\delta > 0$, i.e., alphabets of the form

$$\mathscr{A} = \mathscr{A}_\infty^\delta := \{k\delta : k \in \mathbb{Z}\} \tag{4.3}$$

and their finite versions used in practice:

$$\mathscr{A} = \mathscr{A}_K^\delta := \{\pm k\delta : 0 \le k \le K, k \in \mathbb{Z}\}. \tag{4.4}$$

For example, $\mathscr{A}_1^1 = \{0, \pm 1\}$ is a ternary alphabet. Additionally, we use the following alphabets for sparse GPFQ in Section 4.4.

$$\widetilde{\mathscr{A}} = \mathscr{A}_\infty^{\delta,\lambda} := \{0\} \cup \{\pm(\lambda + k\delta) : k \ge 0, k \in \mathbb{Z}\} \tag{4.5}$$

and

$$\widetilde{\mathscr{A}} = \mathscr{A}_K^{\delta,\lambda} := \{0\} \cup \{\pm(\lambda + k\delta) : 0 \le k \le K, k \in \mathbb{Z}\} \tag{4.6}$$

where $\delta > 0$ denotes the quantization step size and $\lambda > 0$ is a threshold. Moreover, for alphabet $\mathscr{A}$ in (4.3) and (4.4), we define the associated *memoryless scalar quantizer* (MSQ) $\mathscr{Q} : \mathbb{R} \to \mathscr{A}$ by

$$\mathscr{Q}(z) := \operatorname*{arg\,min}_{p \in \mathscr{A}} |z - p| = \begin{cases} \delta \operatorname{sign}(z) \left| \left\lfloor \frac{z}{\delta} + \frac{1}{2} \right\rfloor \right| & \text{if } \mathscr{A} = \mathscr{A}_\infty^\delta, \\ \delta \operatorname{sign}(z) \min\left\{ \left| \left\lfloor \frac{z}{\delta} + \frac{1}{2} \right\rfloor \right|, K \right\} & \text{if } \mathscr{A} = \mathscr{A}_K^\delta. \end{cases} \tag{4.7}$$

Further, the MSQ over $\widetilde{\mathscr{A}}$ in (4.5) and (4.6) is given by

$$\widetilde{\mathscr{Q}}(z) := \begin{cases} 0 & \text{if } |z| \le \lambda, \\ \operatorname*{arg\,min}_{p \in \widetilde{\mathscr{A}}} |z - p| & \text{otherwise,} \end{cases}$$

which is equivalent to

$$\widetilde{Q}(z) = \begin{cases} \mathbb{1}_{\{|z|>\lambda\}} \operatorname{sign}(z)\left(\lambda + \delta\left|\left\lfloor \frac{s_\lambda(z)}{\delta} + \frac{1}{2}\right\rfloor\right|\right) & \text{if } \widetilde{\mathscr{A}} = \mathscr{A}_\infty^{\delta,\lambda}, \\ \mathbb{1}_{\{|z|>\lambda\}} \operatorname{sign}(z)\left(\lambda + \delta \min\left\{\left|\left\lfloor \frac{s_\lambda(z)}{\delta} + \frac{1}{2}\right\rfloor\right|, K\right\}\right) & \text{if } \widetilde{\mathscr{A}} = \mathscr{A}_K^{\delta,\lambda}. \end{cases} \tag{4.8}$$

Here, $s_\lambda(z) := \operatorname{sign}(z)\max\{|z|-\lambda, 0\}$ is the *soft thresholding* function and its counterpart, *hard thresholding* function, is defined by

$$h_\lambda(z) := z\mathbb{1}_{\{|z|>\lambda\}} = \begin{cases} z & \text{if } |z| > \lambda, \\ 0 & \text{otherwise.} \end{cases}$$

Algorithm 4: Using GPFQ to quantize MLPs

Input: A L-layer MLP Φ with weight matrices $W^{(i)} \in \mathbb{R}^{N_{i-1}\times N_i}$, input mini-batches $\{X_i\}_{i=1}^L \subset \mathbb{R}^{m\times N_0}$

for $i = 1$ **to** L **do**
- **Phase I**: Forward propagation
- Generate $X^{(i-1)} = \Phi^{(i-1)}(X_i) \in \mathbb{R}^{m\times N_{i-1}}$ and $\widetilde{X}^{(i-1)} = \widetilde{\Phi}^{(i-1)}(X_i) \in \mathbb{R}^{m\times N_{i-1}}$
- **Phase II**: Parallel quantization for $W^{(i)}$
- **repeat**
- Pick a column (neuron) $w \in \mathbb{R}^{N_{i-1}}$ of $W^{(i)}$ and set $u_0 = 0 \in \mathbb{R}^m$
- **for** $t = 1$ **to** N_{i-1} **do**
  - Implement (4.11) and $u_t = u_{t-1} + w_t X_t^{(i-1)} - q_t \widetilde{X}_t^{(i-1)}$
- **until** All columns of $W^{(i)}$ are quantized
- Obtain quantized i-th layer $Q^{(i)} \in \mathscr{A}^{N_{i-1}\times N_i}$

Output: Quantized neural network $\widetilde{\Phi}$

4.2.2 GPFQ

Given a data set $X \in \mathbb{R}^{m\times N_0}$ with vectorized data stored as rows and a trained neural network Φ with weight matrices $W^{(i)}$, the GPFQ algorithm [22] is a map $W^{(i)} \to Q^{(i)} \in \mathscr{A}^{N_{i-1}\times N_i}$, giving a new quantized neural network $\widetilde{\Phi}$ with $\widetilde{\Phi}(X) \approx \Phi(X)$. The matrices $W^{(1)}, \ldots, W^{(L)}$

are quantized sequentially and in each layer every neuron (a column of $W^{(i)}$) is quantized independently of other neurons, which allows parallel quantization across neurons in a layer.

Thus, GPFQ can be implemented recursively. Let $\Phi^{(i)}$, $\widetilde{\Phi}^{(i)}$ denote the original and quantized neural networks up to layer i respectively. Assume the first $i-1$ layers have been quantized and define $X^{(i-1)} := \Phi^{(i-1)}(X)$, $\widetilde{X}^{(i-1)} := \widetilde{\Phi}^{(i-1)}(X) \in \mathbb{R}^{m\times N_{i-1}}$. Then each neuron $w \in \mathbb{R}^{N_{i-1}}$ in layer i is quantized by constructing $q \in \mathscr{A}^{N_{i-1}}$ such that

$$\widetilde{X}^{(i-1)}q = \sum_{t=1}^{N_{i-1}} q_t \widetilde{X}_t^{(i-1)} \approx \sum_{t=1}^{N_{i-1}} w_t X_t^{(i-1)} = X^{(i-1)}w$$

where $X_t^{(i-1)}$, $\widetilde{X}_t^{(i-1)}$ are the t-th columns of $X^{(i-1)}$, $\widetilde{X}^{(i-1)}$. This is done by selecting q_t, for $t = 1,2,\ldots,N_{i-1}$, so the running sum $\sum_{j=1}^{t} q_j \widetilde{X}_j^{(i-1)}$ tracks its analog $\sum_{j=1}^{t} w_j X_j^{(i-1)}$ as well as possible in an ℓ_2 sense. So,

$$q_t = \arg\min_{p\in\mathscr{A}} \Big\| \sum_{j=1}^{t} w_j X_j^{(i-1)} - \sum_{j=1}^{t-1} q_j \widetilde{X}_j^{(i-1)} - p\widetilde{X}_t^{(i-1)} \Big\|_2^2. \tag{4.9}$$

This is equivalent to the following iteration, which facilitates the analysis of the approximation error:

$$\begin{cases} u_0 = 0 \in \mathbb{R}^m, \\ q_t = \operatorname{argmin}_{p\in\mathscr{A}} \big\| u_{t-1} + w_t X_t^{(i-1)} - p\widetilde{X}_t^{(i-1)} \big\|_2^2, \\ u_t = u_{t-1} + w_t X_t^{(i-1)} - q_t \widetilde{X}_t^{(i-1)}. \end{cases} \tag{4.10}$$

By induction, one can verify that $u_t = \sum_{j=1}^{t} (w_j X_j^{(i-1)} - q_j \widetilde{X}_j^{(i-1)})$ for $t = 0,1,\ldots,N_{i-1}$, and thus $\|u_{N_{i-1}}\|_2 = \|X^{(i-1)}w - \widetilde{X}^{(i-1)}q\|_2$. Moreover, one can derive a closed-form expression of q_t in (4.10) as

$$q_t = \mathscr{Q}\Big(\frac{\langle \widetilde{X}_t^{(i-1)}, u_{t-1} + w_t X_t^{(i-1)} \rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2} \Big), \tag{4.11}$$

which is proved in Lemma 4.6.1. The whole algorithm for quantizing multilayer perceptrons (MLPs) is summarized in Algorithm 4. For the i-th layer, this parallelizable algorithm has run

time complexity $O(mN_{i-1})$ per neuron. Note that in order to quantize convolutional neural networks (CNNs), one can simply vectorize the sliding (convolutional) kernels and unfold, i.e., vectorize, the corresponding image patches. Then, taking the usual inner product on vectors, one can reduce to the case of MLPs, also see Section 4.3.3.

4.3 New Theoretical Results for GPFQ

In this section, we present error bounds for GPFQ with single-layer networks Φ in (4.2) with $L = 1$. Since the error bounds associated with the sparse GPFQ in (4.35) and (4.36) are very similar to the one we have for (4.11), we focus on original GPFQ here and leave the theoretical analysis for sparse GPFQ to Section 4.10.

In the single-layer case, we quantize the weight matrix $W := W^{(1)} \in \mathbb{R}^{N_0 \times N_1}$ and implement (4.10) and (4.11) using $i = 1$. Defining the input data $X := X^{(0)} = \widetilde{X}^{(0)} \in \mathbb{R}^{m \times N_0}$, the iteration can be expressed as

$$\begin{cases} u_0 = 0 \in \mathbb{R}^m, \\ q_t = \mathscr{Q}\big(w_t + \frac{X_t^\top u_{t-1}}{\|X_t\|_2^2}\big), \\ u_t = u_{t-1} + w_t X_t - q_t X_t. \end{cases} \tag{4.12}$$

Moreover, we have $u_t = \sum_{j=1}^t (w_j X_j - q_j X_j)$ for $t = 1, 2 \ldots, N_0$. Clearly, our goal is to control $\|u_t\|_2$. In particular, given $t = N_0$, we recover the ℓ_2 distance between full-precision and quantized pre-activations: $\|u_{N_0}\|_2 = \|Xw - Xq\|_2$.

4.3.1 Bounded Input Data

We start with a quantization error bound where the feature vectors, i.e. columns, of the input data matrix $X \in \mathbb{R}^{m \times N_0}$ are bounded. This general result is then applied to data drawn uniformly from a Euclidean ball, and to Bernoulli random data, showing that the resulting relative square error due to quantization decays linearly with the width N_0 of the network.

Theorem 4.3.1 (Bounded input data)**.** *Suppose that the columns X_t of $X \in \mathbb{R}^{m\times N_0}$ are drawn independently from a probability distribution for which there exists $s \in (0,1)$ and $r > 0$ such that $\|X_t\|_2 \le r$ almost surely, and such that for all unit vector $u \in \mathbb{S}^{m-1}$ we have*

$$\mathbb{E}\frac{\langle X_t, u\rangle^2}{\|X_t\|_2^2} \ge s^2. \tag{4.13}$$

Let $\mathscr{A} = \mathscr{A}_\infty^\delta$ be the alphabet in (4.3) *with step size $\delta > 0$. Let $w \in \mathbb{R}^{N_0}$ be the weights associated with a neuron. Quantizing w using* (4.12)*, we have*

$$\mathrm{P}\Big(\|Xw - Xq\|_2^2 \le \frac{r^2\delta^2}{s^2}\log N_0\Big) \ge 1 - \frac{1}{N_0^2}\Big(2 + \frac{1}{\sqrt{1-s^2}}\Big), \tag{4.14}$$

and

$$\mathrm{P}\Big(\max_{1\le t\le N_0}\|u_t\|_2^2 \le \frac{r^2\delta^2}{s^2}\log N_0\Big) \ge 1 - \frac{1}{N_0}\Big(2 + \frac{1}{\sqrt{1-s^2}}\Big). \tag{4.15}$$

Furthermore, if the activation function $\varphi : \mathbb{R} \to \mathbb{R}$ is ξ-Lipschitz continuous, that is, $|\varphi(x) - \varphi(y)| \le \xi|x-y|$ for all $x, y \in \mathbb{R}$, then we have

$$\mathrm{P}\Big(\|\varphi(Xw) - \varphi(Xq)\|_2^2 \le \frac{r^2\delta^2\xi^2}{s^2}\log N_0\Big) \ge 1 - \frac{1}{N_0^2}\Big(2 + \frac{1}{\sqrt{1-s^2}}\Big). \tag{4.16}$$

Proof. Let $\alpha > 0$ and $\eta > 0$. In the t-th step, by Markov's inequality, one can get

$$\mathrm{P}(\|u_t\|_2^2 \ge \alpha) = \mathrm{P}(e^{\eta\|u_t\|_2^2} \ge e^{\eta\alpha}) \le e^{-\eta\alpha}\mathbb{E}e^{\eta\|u_t\|_2^2}. \tag{4.17}$$

Since $\mathscr{A} = \mathscr{A}_\infty^\delta$ is infinite, applying Lemma 4.6.3 with $q_{\max} = \infty$, we have

$$\|u_t\|_2^2 \le \frac{\delta^2}{4}\|X_t\|_2^2 + (1 - \cos^2\theta_t)\|u_{t-1}\|_2^2 \tag{4.18}$$

where $\theta_t = \angle(X_t, u_{t-1})$ is the angle between X_t and u_{t-1}. This yields

$$\mathbb{E}e^{\eta\|u_t\|_2^2} \leq \mathbb{E}(e^{\frac{\eta\delta^2}{4}\|X_t\|_2^2 e^{\eta\|u_{t-1}\|_2^2(1-\cos^2\theta_t)}}). \tag{4.19}$$

Moreover, observing that $\|X_t\|_2^2 \leq r^2$ a.s., then applying the law of total expectation, Lemma 4.6.4 with $\beta = 1$, and assumption (4.13) sequentially, we obtain

$$\begin{aligned}
\mathbb{E}(e^{\frac{\eta\delta^2}{4}\|X_t\|_2^2 e^{\eta\|u_{t-1}\|_2^2(1-\cos^2\theta_t)}}) &\leq e^{\eta r^2\delta^2/4}\mathbb{E}e^{\eta\|u_{t-1}\|_2^2(1-\cos^2\theta_t)} \\
&= e^{\eta r^2\delta^2/4}\mathbb{E}(\mathbb{E}(e^{\eta\|u_{t-1}\|_2^2(1-\cos^2\theta_t)} \mid \mathscr{F}_{t-1})) \\
&\leq e^{\eta r^2\delta^2/4}\mathbb{E}\Big(-\mathbb{E}(\cos^2\theta_t \mid \mathscr{F}_{t-1})(e^{\eta\|u_{t-1}\|_2^2}-1) + e^{\eta\|u_{t-1}\|_2^2}\Big) \\
&\leq e^{\eta r^2\delta^2/4}\mathbb{E}(-s^2(e^{\eta\|u_{t-1}\|_2^2}-1) + e^{\eta\|u_{t-1}\|_2^2}) \\
&= (1-s^2)e^{\eta r^2\delta^2/4}\mathbb{E}e^{\eta\|u_{t-1}\|_2^2} + s^2 e^{\eta r^2\delta^2/4}
\end{aligned}$$

Hence, for each t, inequality (4.19) becomes

$$\mathbb{E}e^{\eta\|u_t\|_2^2} \leq a\mathbb{E}e^{\eta\|u_{t-1}\|_2^2} + b \tag{4.20}$$

where $a := (1-s^2)e^{\eta r^2\delta^2/4}$ and $b := s^2 e^{\eta r^2\delta^2/4}$. Then, noting that $u_0 = 0$, the following inequality follows from (4.20),

$$\mathbb{E}e^{\eta\|u_t\|_2^2} \leq a^t\mathbb{E}e^{\eta\|u_0\|_2^2} + b(1+a+\ldots+a^{t-1}) = a^t + \frac{b(1-a^t)}{1-a} \leq 1 + \frac{b}{1-a} \tag{4.21}$$

where the last inequality holds provided that $a = (1-s^2)e^{\eta r^2\delta^2/4} < 1$. Since the result above hold for all $\eta > 0$ such that $(1-s^2)e^{\eta r^2\delta^2/4} < 1$, we can choose $\eta = \frac{-2\log(1-s^2)}{r^2\delta^2}$. Then we get

$a = (1-s^2)^{1/2}$ and $b = s^2(1-s^2)^{-1/2}$. It follows from (4.17) and (4.21) that

$$\begin{aligned}
\mathrm{P}(\|u_t\|_2^2 \geq \alpha) &\leq e^{-\eta\alpha}\Big(1+\frac{b}{1-a}\Big) = \exp\Big(\frac{2\alpha\log(1-s^2)}{r^2\delta^2}\Big)\Big(1+\frac{s^2(1-s^2)^{-1/2}}{1-(1-s^2)^{1/2}}\Big) \\
&= \exp\Big(\frac{2\alpha\log(1-s^2)}{r^2\delta^2}\Big)\Big(1+(1-s^2)^{-1/2}(1+(1-s^2)^{1/2})\Big) \\
&= \exp\Big(\frac{2\alpha\log(1-s^2)}{r^2\delta^2}\Big)\Big(2+\frac{1}{\sqrt{1-s^2}}\Big) \\
&\leq \exp\Big(\frac{-2\alpha s^2}{r^2\delta^2}\Big)\Big(2+\frac{1}{\sqrt{1-s^2}}\Big).
\end{aligned}$$

The last inequality can be obtained using the fact $\log(1+x) \leq x$ for all $x > -1$. Picking $\alpha = \frac{r^2\delta^2\log N_0}{s^2}$, we get

$$\mathrm{P}\Big(\|u_t\|_2^2 \geq \frac{r^2\delta^2}{s^2}\log N_0\Big) \leq \frac{1}{N_0^2}\Big(2+\frac{1}{\sqrt{1-s^2}}\Big). \tag{4.22}$$

From (4.22) we can first deduce (4.14), by setting $t = N_0$ and using the fact $u_{N_0} = Xw - Xq$. If the activation function φ is ξ-Lipschitz, then $\|\varphi(Xw) - \varphi(Xq)\|_2 \leq \xi\|Xw - Xq\|_2$ and (4.14) implies (4.16). Moreover, applying a union bound over t to (4.22), one can get (4.15). □

Theorem 4.3.1 makes the simplifying assumption that we use an infinite alphabet, namely $\mathscr{A}_\infty^\delta$. This assumption implies that the argument of the scalar quantizer $\mathscr{Q}$ in (4.12) is trivially bounded by the largest alphabet element which in turn implies that $\Big|w_t + \frac{\langle X_t, u_{t-1}\rangle}{\|X_t\|_2^2} - q_t\Big|$ is bounded by $\delta/2$. This fact is used in inequality (4.18). In order to use a finite alphabet $\mathscr{A}_K^\delta$ instead and still have (4.18), the argument of the quantizer must be bounded by $K\delta$. Corollary 4.3.2 shows that with high probability this is indeed the case and Remark 4.3.3 shows that a finite alphabet with $K \approx \log(N_0)$ suffices for our purposes.

Corollary 4.3.2. *Let $\gamma > 0$. Under the conditions of Theorem 4.3.1, suppose that there exist*

constants $c_1, c_2 > 0$ so that the columns X_t of $X \in \mathbb{R}^{m\times N_0}$ also satisfy

$$\mathrm{P}\Big(\Big|\frac{\langle X_t, u\rangle}{\|X_t\|_2^2}\Big| \geq \frac{s\sqrt{\gamma \log N_0}}{r}\Big) \leq c_1 N_0^{-c_2\gamma} \tag{4.23}$$

for any unit vector $u \in \mathbb{S}^{m-1}$. Then

$$\max_{1\leq t\leq N_0}\Big|w_t + \frac{\langle X_t, u_{t-1}\rangle}{\|X_t\|_2^2}\Big| \leq \|w\|_\infty + \gamma^{\frac{1}{2}}\delta\log N_0 \quad \textit{and} \quad \max_{1\leq t\leq N_0}\|u_t\|_2^2 \leq \frac{r^2\delta^2}{s^2}\log N_0. \tag{4.24}$$

hold with probability at least $1 - \frac{1}{N_0}\Big(2 + \frac{1}{\sqrt{1-s^2}}\Big) - \frac{c_1}{N_0^{c_2\gamma-1}}$.

Proof. Consider the t-th iteration of (4.12) and let $\mathscr{E}_{t-1}$ be the event $\|u_{t-1}\|_2^2 \leq \frac{r^2\delta^2}{s^2}\log N_0$. By Theorem 4.3.1, we have

$$\mathrm{P}(\mathscr{E}_{t-1}) \geq 1 - \frac{1}{N_0^2}\Big(2 + \frac{1}{\sqrt{1-s^2}}\Big). \tag{4.25}$$

Conditioning on $\mathscr{E}_{t-1}$ and applying (4.23), we have

$$\begin{aligned}
\mathrm{P}\Big(\Big|w_t + \frac{\langle X_t, u_{t-1}\rangle}{\|X_t\|_2^2}\Big| > \|w\|_\infty + \gamma^{\frac{1}{2}}\delta\log N_0 \mid \mathscr{E}_{t-1}\Big) &\leq \mathrm{P}\Big(\Big|\frac{\langle X_t, u_{t-1}\rangle}{\|X_t\|_2^2\|u_{t-1}\|_2}\Big| \geq \frac{\gamma^{\frac{1}{2}}\delta\log N_0}{\|u_{t-1}\|_2} \mid \mathscr{E}_{t-1}\Big) \\
&\leq \mathrm{P}\Big(\Big|\frac{\langle X_t, u_{t-1}\rangle}{\|X_t\|_2^2\|u_{t-1}\|_2}\Big| \geq \frac{s\sqrt{\gamma\log N_0}}{r} \mid \mathscr{E}_{t-1}\Big) \\
&\leq c_1 N_0^{-c_2\gamma}.
\end{aligned} \tag{4.26}$$

Combining (4.25) and (4.26), we obtain

$$\mathrm{P}\Big(\Big|w_t + \frac{\langle X_t, u_{t-1}\rangle}{\|X_t\|_2^2}\Big| \leq \|w\|_\infty + \gamma^{\frac{1}{2}}\delta\log N_0, \mathscr{E}_{t-1}\Big) \geq 1 - \frac{1}{N_0^2}\Big(2 + \frac{1}{\sqrt{1-s^2}}\Big) - \frac{c_1}{N_0^{c_2\gamma}}.$$

Taking a union bound over t, we obtain the desired result. □

Remark 4.3.3 (Finite alphabets suffice)**.** *By* (4.24) *and the definition of q_t in* (4.12)*, we see that with high probability only the elements q of the alphabet with $|q| \leq \|w\|_\infty + \gamma^{1/2}\delta\log N_0 + \delta$ are used. So, on this high probability event, we can simply replace $\mathscr{A}_\infty^\delta$ by $\mathscr{A}_K^\delta$ where the largest*

element $q_{\max} = K\delta$ *satisfies* $q_{\max} = K\delta \geq \|w\|_\infty + \gamma^{1/2}\delta \log N_0$.

Next, we illustrate how Corollary 4.3.2 can be applied to obtain error bounds associated with uniformly distributed and Bernoulli distributed inputs.

Uniformly Distributed Data

Let $B_r \subset \mathbb{R}^m$ be the closed ball with center 0 and radius $r > 0$. Suppose that columns X_t of $X \in \mathbb{R}^{m\times N_0}$ are drawn i.i.d. from $\mathrm{Unif}(B_r)$. Then we can represent X_t as $X_t = rU^{\frac{1}{m}}Z$ where $U \sim \mathrm{Unif}([0,1])$ and $Z \sim \mathrm{Unif}(\mathbb{S}^{m-1})$ are independent, so that $\|X_t\|_2 = rU^{\frac{1}{m}}$ and $X_t/\|X_t\|_2 = Z$. Fix a unit vector $u \in \mathbb{S}^{m-1}$ and $\gamma > 0$. Since Z is rotation invariant, we have $\mathbb{E}\frac{\langle X_t,u\rangle^2}{\|X_t\|_2^2} = \mathbb{E}\langle Z,u\rangle^2 = \mathbb{E}\langle Z,e_1\rangle^2 = \mathbb{E}Z_1^2 = \frac{1}{m}$. The last equality holds because $\|Z\|_2 = 1$ and $\mathbb{E}Z_1^2 = \mathbb{E}Z_2^2 = \ldots = \mathbb{E}Z_m^2 = \frac{1}{m}\mathbb{E}\Big(\sum_{i=1}^m Z_i^2\Big) = \frac{1}{m}$. Additionally, we have $\mathrm{P}(\|X_t\|_2 \geq \frac{r}{2}) = \mathrm{P}(U^{\frac{1}{m}} \geq \frac{1}{2}) = 1 - \frac{1}{2^m}$ and

$$\begin{aligned}
&\mathrm{P}\Big(\Big|\frac{\langle X_t,u\rangle}{\|X_t\|_2^2}\Big| \geq \frac{1}{r}\sqrt{\frac{\gamma\log N_0}{m}} \mid \|X_t\|_2 \geq \frac{r}{2}\Big) = \mathrm{P}\Big(|\langle Z,u\rangle| \geq \frac{\|X_t\|_2}{r}\sqrt{\frac{\gamma\log N_0}{m}} \mid \|X_t\|_2 \geq \frac{r}{2}\Big) \\
&\leq \mathrm{P}\Big(|\langle Z,u\rangle| \geq \frac{1}{2}\sqrt{\frac{\gamma\log N_0}{m}}\Big) = \mathrm{P}\Big(|Z_1| \geq \frac{1}{2}\sqrt{\frac{\gamma\log N_0}{m}}\Big) \leq 4\exp(-c\gamma\log N_0).
\end{aligned}$$

In the last step, we used Theorem 3.4.6 in [29]. It follows that $\mathrm{P}\Big(\Big|\frac{\langle X_t,u\rangle}{\|X_t\|_2^2}\Big| \geq \frac{1}{r}\sqrt{\frac{\gamma\log N_0}{m}}\Big) \leq 4\exp(-c\gamma\log N_0) + \frac{1}{2^m}$. If $m \geq c\gamma\log_2 N_0$, then we have $4\exp(-c\gamma\log N_0) + \frac{1}{2^m} \leq 5N_0^{-c\gamma}$ and thus (4.13) and (4.23) hold with $s^2 = \frac{1}{m}$. Choosing $\gamma = 2c^{-1}$ and alphabet $\mathscr{A}_K^\delta$ with $K \geq \delta^{-1}\|w\|_\infty + \gamma^{\frac{1}{2}}\log N_0$, Corollary 4.3.2 implies that, with high probability

$$\|Xw - Xq\|_2^2 \lesssim mr^2\delta^2\log N_0. \tag{4.27}$$

Moreover, $\mathbb{E}\|X_t\|_2^2 = r^2\mathbb{E}U^{\frac{2}{m}} = \frac{mr^2}{m+2}$. Then $\mathbb{E}(X^\top X) = \mathbb{E}\|X_1\|_2^2 I_{N_0} = \frac{mr^2}{m+2}I_{N_0}$ and thus $\mathbb{E}\|Xw\|_2^2 = w^\top\mathbb{E}(X^\top X)w = \frac{mr^2}{m+2}\|w\|_2^2$. If the weight vector $w \in \mathbb{R}^{N_0}$ is *generic* in the sense that $\|w\|_2^2 \gtrsim N_0$, then

$$\mathbb{E}\|Xw\|_2^2 \gtrsim \frac{mN_0r^2}{m+2}. \tag{4.28}$$

Combining (4.27) with (4.28), the relative error satisfies $\frac{\|Xw-Xq\|_2^2}{\|Xw\|_2^2} \lesssim \frac{m\delta^2 \log N_0}{N_0}$.

Data from a Symmetric Bernoulli Distribution

We say that a random vector $Z = (Z_1, Z_2, \ldots, Z_m)$ is *symmetric Bernoulli* if the coordinates Z_i are independent and $\mathrm{P}(Z_i = 1) = \mathrm{P}(Z_i = -1) = \frac{1}{2}$. Now assume that columns X_t of $X \in \mathbb{R}^{m \times N_0}$ are independent and subject to symmetric Bernoulli distribution. Clearly, $\|X_t\|_2 = \sqrt{m}$. Let $u \in \mathbb{R}^m$ be a unit vector and $\gamma > 0$. Then $\mathbb{E}\frac{\langle X_t, u\rangle^2}{\|X_t\|_2^2} = \frac{u^\top \mathbb{E}(X_t X_t^\top) u}{m} = \frac{\|u\|_2^2}{m} = \frac{1}{m}$ and $\mathrm{P}\left(\left|\frac{\langle X_t, u\rangle}{\|X_t\|_2^2}\right| \geq \frac{\sqrt{\gamma \log N_0}}{m}\right) = \mathrm{P}\left(|\langle X_t, u\rangle| \geq \sqrt{\gamma \log N_0}\right) \leq 2\exp(-\frac{1}{2}\gamma \log N_0) = 2N_0^{-\frac{\gamma}{2}}$ by Hoeffding's inequality. Thus (4.13) and (4.23) hold with $s^2 = \frac{1}{m}$. Picking $\gamma = 4$ and alphabet $\mathscr{A}_K^\delta$ with $K \geq \delta^{-1}\|w\|_\infty + \gamma^{\frac{1}{2}} \log N_0$, Corollary 4.3.2 implies that

$$\|Xw - Xq\|_2^2 \leq m^2 \delta^2 \log N_0 \tag{4.29}$$

holds with high probability. Again, a generic $w \in \mathbb{R}^{N_0}$ with $\|w\|_2^2 \gtrsim N_0$ satisfies $\mathbb{E}\|Xw\|_2^2 = w^\top \mathbb{E}(X^\top X) w = m\|w\|_2^2 \gtrsim mN_0$ and therefore $\frac{\|Xw-Xq\|_2^2}{\|Xw\|_2^2} \lesssim \frac{m\delta^2 \log N_0}{N_0}$.

4.3.2 Gaussian Clusters

Here, we consider data drawn from Gaussian clusters, which unlike the previously considered models, are unbounded. One reason for considering Gaussian clusters is that they are a reasonable model for the activations in deeper layers of networks designed for classification. Specifically, suppose our samples are drawn from d normally distributed clusters $\mathscr{K}_i := \mathscr{N}(z^{(i)}, \sigma^2 I_{N_0})$ with fixed centers $z^{(i)} \in \mathbb{R}^{N_0}$ and $\sigma > 0$. Suppose, for simplicity, that we independently draw n samples from each cluster and vertically stack them in order as rows of X (this ordering does not affect our results in Theorem 4.3.4). Let $m := nd$. So, for $1 \leq i \leq d$, the row indices of X ranging from $(i-1)n+1$ to in come from cluster $\mathscr{K}_i$. Then the t-th column of X is of the form

$$X_t = [Y_t^{(1)}, Y_t^{(2)}, \ldots, Y_t^{(d)}]^\top \in \mathbb{R}^m \tag{4.30}$$

where $Y_t^{(i)} \sim \mathcal{N}(z_t^{(i)} \mathbb{1}_n, \sigma^2 I_n)$.

Theorem 4.3.4 (Gaussian clusters)**.** *Let $X \in \mathbb{R}^{m\times N_0}$ be as in* (4.30) *and let $\mathscr{A} = \mathscr{A}_\infty^\delta$ be as in* (4.3)*, with step size $\delta > 0$. Let $p \in \mathbb{N}$, $J := 1 + (d\sigma^2)^{-1} \max_{1\le t\le N_0} \sum_{i=1}^d (z_t^{(i)})^2$, and $w \in \mathbb{R}^{N_0}$ be the weights associated with a neuron. Quantizing w using* (4.12)*, we have*

$$\mathrm{P}\Big(\|Xw - Xq\|_2^2 \ge 4pm^2J^2\delta^2\sigma^2 \log N_0\Big) \le \frac{7\sqrt{mJ}}{N_0^p}, \quad \textit{and}$$

$$\mathrm{P}\Big(\max_{1\le t\le N_0} \|u_t\|_2^2 \ge 4pm^2J^2\delta^2\sigma^2 \log N_0\Big) \le \frac{7\sqrt{mJ}}{N_0^{p-1}}.$$

If the activation function φ is ξ-Lipschitz continuous, then

$$\mathrm{P}\Big(\|\varphi(Xw) - \varphi(Xq)\|_2^2 \ge 4pm^2J^2\xi^2\delta^2\sigma^2 \log N_0\Big) \le \frac{7\sqrt{mJ}}{N_0^p}.$$

Moreover, similar to Corollary 4.3.2, we show that, with high probability, the same error bounds hold using finite alphabets $\mathscr{A}_K^\delta$.

Corollary 4.3.5. *Under the conditions of Theorem 4.3.4, suppose that $X \in \mathbb{R}^{m\times N_0}$ also satisfies $J \le 1 + \frac{\log N_0}{36m}$ and $m \ge \max\{1, \frac{2}{c_1}\} \log N_0$ where c_1 is an absolute constant defined in Lemma 4.9.3. Then*

$$\max_{1\le t\le N_0} \Big| w_t + \frac{\langle X_t, u_{t-1}\rangle}{\|X_t\|_2^2} \Big| \le \|w\|_\infty + 8\delta J\sqrt{p}\log N_0 \quad \textit{and} \quad \max_{1\le t\le N_0} \|u_{t-1}\|_2^2 \le 4pm^2J^2\delta^2\sigma^2 \log N_0 \tag{4.31}$$

holds with probability at least $1 - \frac{7\sqrt{mJ}}{N_0^{p-1}} - \frac{3}{N_0}$.

According to (4.31), it is sufficient to quantize w using a finite alphabet $\mathscr{A}_K^\delta$ where K satisfies

$$\delta^{-1}\|w\|_\infty + 8J\sqrt{p}\log N_0 \le \delta^{-1}\|w\|_\infty + 9\sqrt{p}\log N_0 \le K.$$

In the first step, we used $J \le 1 + \frac{\log N_0}{36m} \le \frac{9}{8}$. The proof of Theorem 4.3.4 and Corollary 4.3.5 can

be found in Section 4.9.1 and Section 4.9.2 respectively.

Normally Distributed Data

As a special case of (4.30), let $X \in \mathbb{R}^{m\times N_0}$ be a Gaussian matrix with $X_{ij} \overset{i.i.d.}{\sim} \mathcal{N}(0,\sigma^2)$ corresponding to $d = 1$, $n = m$, and $z^{(1)} = 0$. Theorem 4.3.4 implies that $J = 1$ and

$$\mathrm{P}\Big(\|Xw - Xq\|_2^2 \geq 4pm^2\delta^2\sigma^2 \log N_0\Big) \leq \frac{7\sqrt{m}}{N_0^p}. \tag{4.32}$$

Further, suppose that $w \in \mathbb{R}^{N_0}$ is generic, i.e. $\|w\|_2^2 \gtrsim N_0$. In this case, $\mathbb{E}\|Xw\|_2^2 = m\sigma^2\|w\|_2^2 \gtrsim m\sigma^2 N_0$. So, with high probability, the relative error in our quantization satisfies

$$\frac{\|Xw - Xq\|_2^2}{\|Xw\|_2^2} \lesssim \frac{m\delta^2 \log N_0}{N_0}. \tag{4.33}$$

Thus, here again, the relative square error for quantizing a single-layer MLP decays linearly (up to a log factor) in the number of neurons N_0. Note that (4.33), for ternary alphabets, is the main result given by [22], which we now obtain as a special case of Theorem 4.3.4.

Remark 4.3.6. *In Section 4.3.1 and Section 4.3.2, we have shown that if the columns of $X \in \mathbb{R}^{m\times N_0}$ are drawn from proper distributions, then the relative error for quantization is small when $m \ll N_0$. Now consider the case where the feature vectors $\{X_t\}_{t=1}^{N_0}$ live in a l-dimensional subspace with $l < m$. In this case, $X = VF$ where $V \in \mathbb{R}^{m\times l}$ satisfies $V^\top V = I$, and the columns F_t of $F \in \mathbb{R}^{l\times N_0}$ are drawn i.i.d. from a distribution $\mathscr{P}$. Suppose, for example, that $\mathscr{P} = \mathrm{Unif}(B_r)$. Due to $X = VF$, one can express any unit vector in the range of X as $u = Vv$ with $v \in \mathbb{R}^l$. Then we have $1 = \|u\|_2 = \|Vv\|_2 = \|v\|_2$, $\|X_t\|_2 = \|VF_t\|_2 = \|F_t\|_2 \leq r$, and $\mathbb{E}\frac{\langle X_t,u\rangle^2}{\|X_t\|_2^2} = \mathbb{E}\frac{\langle VF_t,Vv\rangle^2}{\|VF_t\|_2^2} = \mathbb{E}\frac{\langle F_t,v\rangle^2}{\|F_t\|_2^2} = l^{-1}$ by our assumption for $\mathscr{P}$. Because u_t in Theorem 4.3.1 is a linear combination of X_j, the proof of Theorem 4.3.1 remains unchanged if* (4.13) *holds for all unit vectors u in the range of X. It follows that Theorem 4.3.1 holds for X with $s^2 = l^{-1}$ and thus the relative error for quantizing the data in a l-dimensional subspace is improved*

to $\frac{\|Xw-Xq\|_2^2}{\|Xw\|_2^2} \leq \frac{l\delta^2 \log N_0}{N_0}$. *Applying a similar argument to* $\mathscr{P}$ *representing either a symmetric Bernoulli distribution or Gaussian distribution, one can replace m in their corresponding relative errors by l. In short, the relative error depends not on the number of training samples m but on the intrinsic dimension of the features l.*

4.3.3 Convolutional Neural Networks

In this section, we derive error bounds for single-layer CNNs. Let $Z \in \mathbb{R}^{B\times C_{\text{in}}\times S_1\times S_2}$ be a mini-batch of images with batch size B, input channels C_{in}, height S_1, and width S_2. Suppose that all entries of Z are i.i.d. drawn from $\mathscr{N}(0,1)$ and suppose we have C_{out} convolutional kernels $\{w_i\}_{i=1}^{C_{\text{out}}} \subseteq \mathbb{R}^{C_{\text{in}}\times k_1\times k_2}$. Let these kernels "slide" over Z with fixed stride (k_1,k_2) such that sliding local blocks generated by moving w_i on Z are disjoint. Additionally, if T is the number of randomly selected sliding local blocks (in $\mathbb{R}^{C_{\text{in}}\times k_1\times k_2}$) from each image, then one can vectorize all BT local blocks and stack them together to obtain a single data matrix $X \in \mathbb{R}^{BT\times C_{\text{in}}k_1k_2}$. Moreover, each kernel w_i can be viewed as a column vector in $\mathbb{R}^{C_{\text{in}}k_1k_2}$ and thus $W = [w_1, w_2, \ldots, w_{C_{\text{out}}}] \in \mathbb{R}^{C_{\text{in}}k_1k_2\times C_{\text{out}}}$ is the weight matrix to be quantized. Thus, we need to convert W to $Q = [q_1, q_2, \ldots, q_{C_{\text{out}}}] \in \mathscr{A}^{C_{\text{in}}k_1k_2\times C_{\text{out}}}$ with $XQ \approx XW$, as before. Since extracted local blocks from Z are disjoint, columns of X are independent and subject to $\mathscr{N}(0, I_{BT})$. Hence, one can apply (4.32) with $m = BT$, $N_0 = C_{\text{in}}k_1k_2$, $\sigma = 1$, and any $p \in \mathbb{N}$. Specifically, for $1 \leq i \leq C_{\text{out}}$, we get $\mathrm{P}\Big(\|Xw_i - Xq_i\|_2^2 \geq 4pB^2T^2\delta^2 \log(C_{\text{in}}k_1k_2)\Big) \lesssim \frac{\sqrt{BT}}{(C_{\text{in}}k_1k_2)^p}$. By a union bound, $\mathrm{P}\Big(\max_{1\leq i\leq C_{\text{out}}} \|Xw_i - Xq_i\|_2^2 \geq 4pB^2T^2\delta^2 \log(C_{\text{in}}k_1k_2)\Big) \lesssim \frac{C_{\text{out}}\sqrt{BT}}{(C_{\text{in}}k_1k_2)^p}$.

4.4 Sparse GPFQ and Error Analysis

Having extended the results pertaining to GPFQ to cover multiple distributions of the input data, as well as general alphabets, we now propose modifications to produce quantized weights that are also sparse, i.e., that have a large fraction of coefficients being 0. Our sparse quantization schemes result from adding a regularization term to (4.10). Specifically, in order to

generate sparse $q \in \mathscr{A}^{N_{i-1}}$, we compute q_t via

$$q_t = \arg\min_{p \in \mathscr{A}} \left(\frac{1}{2} \left\| u_{t-1} + w_t X_t^{(i-1)} - p \widetilde{X}_t^{(i-1)} \right\|_2^2 + \lambda |p| \|\widetilde{X}_t^{(i-1)}\|_2^2 \right) \tag{4.34}$$

where $\lambda > 0$ is a regularization parameter. Conveniently, Lemma 4.6.2 shows that the solution of (4.34) is given by

$$q_t = \mathscr{Q} \circ s_\lambda \left(\frac{\langle \widetilde{X}_t^{(i-1)}, u_{t-1} + w_t X_t^{(i-1)} \rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2} \right) \tag{4.35}$$

where s_λ denotes soft thresholding. It is then natural to consider a variant of (4.35) replacing s_λ with hard thresholding, h_λ. Since $h_\lambda(z)$ has jump discontinuities at $z = \pm\lambda$, the corresponding alphabet and quantizer should be adapted to this change. Thus, we use $\widetilde{\mathscr{Q}}(z)$ over $\widetilde{\mathscr{A}}$ as in (4.8) and $q_t \in \widetilde{\mathscr{A}}$ is obtained via

$$q_t = \widetilde{\mathscr{Q}} \circ h_\lambda \left(\frac{\langle \widetilde{X}_t^{(i-1)}, u_{t-1} + w_t X_t^{(i-1)} \rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2} \right). \tag{4.36}$$

In both cases, we update the error vector via $u_t = u_{t-1} + w_t X_t^{(i-1)} - q_t \widetilde{X}_t^{(i-1)}$, as before. In summary, for quantizing a single-layer network, similar to (4.12) the two sparse GPFQ schemes related to soft and hard thresholding are given by

$$\begin{cases} u_0 = 0 \in \mathbb{R}^m, \\ q_t = \mathscr{Q} \circ s_\lambda \left(w_t + \frac{X_t^\top u_{t-1}}{\|X_t\|_2^2} \right), \\ u_t = u_{t-1} + w_t X_t - q_t X_t. \end{cases} \tag{4.37}$$

$$\begin{cases} u_0 = 0 \in \mathbb{R}^m, \\ q_t = \widetilde{\mathscr{Q}} \circ h_\lambda \left(w_t + \frac{X_t^\top u_{t-1}}{\|X_t\|_2^2} \right), \\ u_t = u_{t-1} + w_t X_t - q_t X_t. \end{cases} \tag{4.38}$$

Interesting, with these sparsity promoting modifications, one can prove similar error bounds to GPFQ. To illustrate with bounded or Gaussian clustered data, we show that sparse GPFQ admits similar error bounds as in Theorem 4.3.1 and Theorem 4.3.4. The following results are proved in Section 4.10.

Theorem 4.4.1 (Sparse GPFQ with bounded input data)**.** *Under the conditions of Theorem 4.3.1, we have the following.*

(*a*) *Quantizing w using* (4.37) *with the alphabet* $\mathscr{A}$ *in* (4.3)*, we have*

$$\mathrm{P}\Big(\|Xw-Xq\|_2^2 \leq \frac{r^2(2\lambda+\delta)^2}{s^2}\log N_0\Big) \geq 1-\frac{1}{N_0^2}\Big(2+\frac{1}{\sqrt{1-s^2}}\Big).$$

(*b*) *Quantizing w using* (4.38) *with the alphabet* $\widetilde{\mathscr{A}}$ *in* (4.5)*, we have*

$$\mathrm{P}\Big(\|Xw-Xq\|_2^2 \leq \frac{r^2\max\{2\lambda,\delta\}^2}{s^2}\log N_0\Big) \geq 1-\frac{1}{N_0^2}\Big(2+\frac{1}{\sqrt{1-s^2}}\Big).$$

Theorem 4.4.2 (Sparse GPFQ for Gaussian clusters)**.** *Under the assumptions of Theorem 4.3.4, the followings inequalities hold.*

(*a*) *Quantizing w using* (4.37) *with the alphabet* $\mathscr{A}$ *in* (4.3)*, we have*

$$\mathrm{P}\Big(\|Xw-Xq\|_2^2 \geq 4pm^2J^2(2\lambda+\delta)^2\sigma^2\log N_0\Big) \leq \frac{7\sqrt{mJ}}{N_0^p}.$$

(*b*) *Quantizing w using* (4.38) *with the alphabet* $\widetilde{\mathscr{A}}$ *in* (4.5)*, we have*

$$\mathrm{P}\Big(\|Xw-Xq\|_2^2 \geq 4pm^2J^2\max\{2\lambda,\delta\}^2\sigma^2\log N_0\Big) \leq \frac{7\sqrt{mJ}}{N_0^p}.$$

Note that the sparsity regularization term λ only appears in the error bounds, making them slightly worse than those where no sparsity is enforced. In Section 4.5.2, we will numerically explore the impact of λ on the sparsity and accuracy of quantized neural networks.

4.5 Experiments

To evaluate the performance of our method and compare it with the approaches reviewed in Section 4.1.1, we test our modified GPFQ on the ImageNet classification task [1]. In particular, we focus on ILSVRC-2012 [5], a 1000-category dataset with over 1.2 million training images and 50 thousand validation images. All images in ILSVRC-2012 are preprocessed in a standard

[1] Our code for experiments is available: https : //github.com/YixuanSeanZhou/Quantized_Neural_Nets.git

manner before they are fed into neural networks: we resize each image to 256×256 and use the normalized 224×224 center crop. The evaluation metrics we use are top-1 and top-5 accuracy of the quantized models on the validation dataset.

4.5.1 Experimental Setup

For reproducibility and fairness of comparison, we use the pretrained 32-bit floating point neural networks provided by torchvision[2] in PyTorch [25]. We test several well-known neural network architectures including: AlexNet [17], VGG-16 [26], GoogLeNet [27], ResNet-18, ResNet-50 [13], and EfficeintNet-B1 [28]. In the following experiments, we will focus on quantizing the weights of fully-connected and convolutional layers of the above architectures, as our theory applies specifically to these types of layers[3].

Let $b \in \mathbb{N}$ denote the number of bits used for quantization. Here, we fix b for all the layers. In our experiments with GPFQ, we adopt the midtread alphabets $\mathscr{A}_K^\delta$ in (4.4) with

$$K = 2^{b-1}, \quad \delta = \frac{R}{2^{b-1}}, \tag{4.39}$$

where $R > 0$ is a hyper-paramter. Indeed, according to (4.4), $\mathscr{A}_K^\delta$ is symmetric with maximal element $q_{\max} = K\delta = R$. Since b is fixed, all that remains is to select R in (4.39) based on the distribution of weights. To that end, suppose we are quantizing the i-th layer of a neural network with weight matrix $W^{(i)} \in \mathbb{R}^{N_{i-1} \times N_i}$. Then, Theorem 4.3.1 and Theorem 4.3.4 require that $R = q_{\max} \geq \max_{k,j} |W_{k,j}^{(i)}|$, and yield error bounds that favor a smaller step size $\delta \propto R$. In practice, however, the weights may have outliers with large magnitudes, which would entail unnecessarily using a large R. Thus, rather than choosing $R = \max_{k,j} |W_{k,j}^{(i)}|$, we will consider the average infinity norm of weights across all neurons w, i.e. columns of $W^{(i)}$. That is

[2] https://pytorch.org/vision/stable/models.html

[3] Batch normalization layers, while not explicitly covered by our methods in the preceeding sections, are easy to handle. Indeed, in Section 4.7, we show that our approach can effectively quantize batch normalization layers by merging them with their preceding convolutional layers before quantization, and we demonstrate experimentally that this does not negatively impact performance.

$R \propto \frac{1}{N_i}\sum_{1\le j\le N_i}\|W_j^{(i)}\|_\infty$. Then, by (4.39), the step size used for quantizing the i-th layer is given by

$$\delta^{(i)} := \frac{C}{2^{b-1}N_i}\sum_{1\le j\le N_i}\|W_j^{(i)}\|_\infty. \tag{4.40}$$

Here, $C \ge 1$ is independent of i and fixed across layers, batch-sizes, and bit widths. To obtain a good choice of C, we perform a grid search with cross-validation over the interval $[1,2]$, albeit on a small batch size $m \le 128$. So the tuning of C takes very little time compared to the quantization with the full training data. Note that the tuning and quantization scale linearly in the size of the data set and the number of parameters of the network. This means that this entire process's computational complexity is dominated by the original training of the network and there is no problem with its scaling to large networks. Moreover, by choosing the maximal element in our alphabet, i.e. $q_{\max} = 2^{b-1}\delta^{(i)}$, to be a constant $C \in [1,2]$ times the average ℓ_∞ norm of all the neurons, we are selecting a number that is effectively larger than most of the weights and thereby corresponding perfectly with the theory for most of the neurons. For the remaining neurons, the vast majority of the weights will be below this threshold, and only the outlier weights, in general, will exceed it. In Section 4.8, we present a theoretical analysis of the expected error when a few weights exceed $q_{\max}$. We not only show that the proposed algorithm is still effective in this scenario, but also that in some cases, it may be beneficial to choose δ small enough such that some weights exceed $q_{\max}$. The analysis in Section 4.8 is consistent with, and helps explain the experimental results in this section. Further, we comment that a more thorough search for an optimal C depending on these individual parameters, e.g. b, may improve performance.

Table 4.1. Top-1/Top-5 accuracy drop using $b = 5$ bits.

Model	C	m	Acc Drop (%)	Model	C	m	Acc Drop (%)
AlexNet	1.1	2048	0.85/0.33	GoogLeNet	1.41	2048	0.60/0.46
VGG-16	1.0	512	0.63/0.32	EfficientNet-B1	1.6	2048	0.45/0.18
ResNet-18	1.16	4096	0.49/0.23	ResNet-50	1.81	2048	0.62/0.11

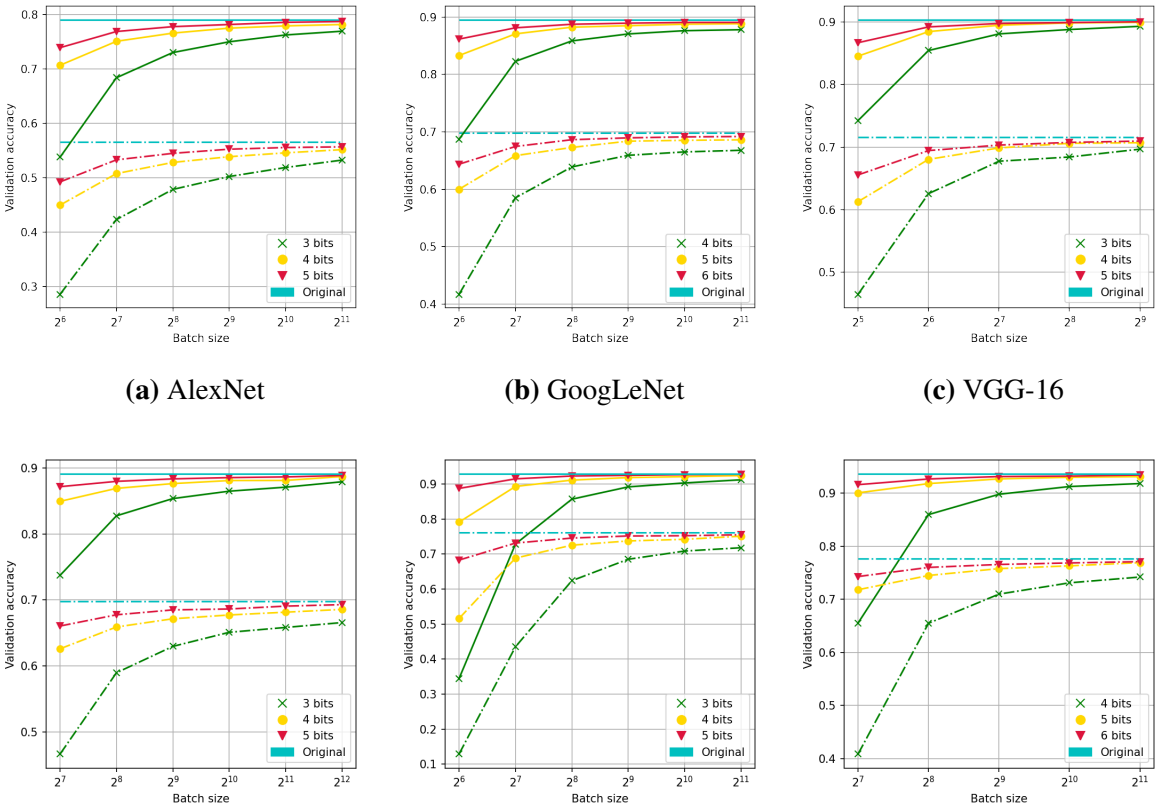
As mentioned in Section 4.3.3, we introduce a sampling probability $p \in (0,1]$, associated

with GPFQ for convolutional layers. This is motivated, in part, by decreasing the computational cost associated with quantizing such layers. Indeed, a batched input tensor of a convolutional layer can be unfolded as a stack of vectorized sliding local blocks, i.e., a matrix. Since, additionally, the kernel can be reshaped into a column vector, matrix-vector multiplication followed by reshaping gives the output of this convolutional layer. On the other hand, due to potentially large overlaps between sliding blocks, the associated matrices have large row size and thus the computational complexity is high. To accelerate our computations, we extract the data used for quantization by setting the stride (which defines the step size of the kernel when sliding through the image) equal to the kernel size and choosing $p = 0.25$. This choice gives a good trade-off between accuracy and computational complexity, which both increase with p. Recall that the batch size $m \in \mathbb{N}$ denotes the number of samples used for quantizing each layer of a neural network. In all experiments, b is chosen from $\{3,4,5,6\}$.

4.5.2 Results on ImageNet

Impact of b and m

The first experiment is designed to explore the effect of the batch size m, as well as bit-width b, on the accuracy of the quantized models. We compute the validation accuracy of quantized networks with respect to different choices of b and m. In particular, Table 4.1 shows that, using $b = 5$ bits, all quantized models achieve less than 1% loss in top-1 and top-5 accuracy. Moreover, we illustrate the relationship between the quantization accuracy and the batch size m in Figure 4.1, where the horizontal lines in cyan, obtained directly from the original validation accuracy of unquantized models, are used for comparison against our quantization method. We observe that (1) all curves with distinct b quickly approach an accuracy ceiling while curves with high b eventually reach a higher ceiling; (2) Quantization with $b \geq 4$ attains near-original model performance with sufficiently large m; (3) one can expect to obtain higher quantization accuracy by taking larger m but the extra improvement that results from increasing the batch size rapidly diminishes.



(a) AlexNet **(b)** GoogLeNet **(c)** VGG-16

(d) ResNet-18 **(e)** ResNet-50 **(f)** EfficientNet-B1

Figure 4.1. Top-1 (dashed lines) and Top-5 (solid lines) accuracy for original and quantized models on ImageNet.

Comparisons with Baselines

Next, we compare GPFQ against other post-training quantization schemes discussed in Section 4.1.1 on various architectures. We note, however, that for a fixed architecture each post-training quantization method starts with a potentially different set of parameters (weights and biases), and these parameters are not available to us. As such, we simply report other methods' accuracies as they appear in their associated papers. Due to this, a perfect comparison between methods is not possible. Another factor that impacts the comparison is that following DoReFa-Net [36], many baseline quantization schemes [34, 14, 19] leave the first and the last layers of DNNs unquantized to alleviate accuracy degradation. On the other hand, we quantize *all* layers of the model. Table 4.2 displays the number of bits and the method used to quantize

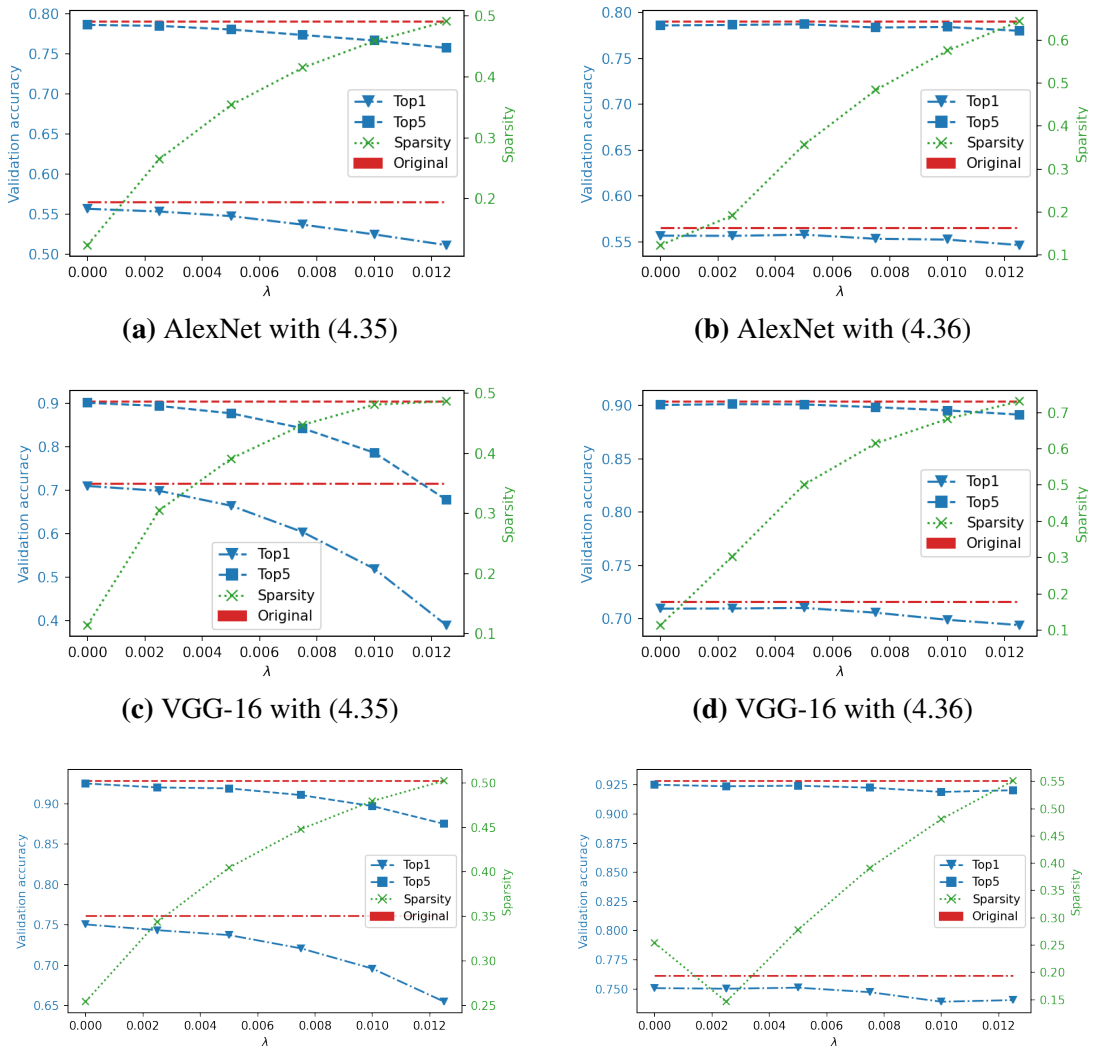
each network. It also contains the accuracy of quantized and full-precision models respectively, as well as their difference, i.e. accuracy drop. We report the results of GPFQ (without the † superscript) for all models with $b = 3, 4, 5$. The important observation here is that our method is competitive across architectures and bit-widths, and shows the best performance on a number of them.

Table 4.2. ImageNet Top-1 accuracy with weight quantization.

Model	Bits	Method	Quant Acc (%)	Ref Acc (%)	Acc Drop (%)
Alexnet	3	GPFQ (Ours)	53.22	56.52	3.30
		GPFQ (Ours)†	54.77	56.52	1.75
	4	OMSE[3]	55.52	56.62	1.10
		GPFQ (Ours)	55.15	56.52	1.37
		GPFQ (Ours)†	55.51	56.52	1.01
	5	GPFQ (Ours)	55.67	56.52	0.85
		GPFQ (Ours)†	55.94	56.52	0.58
	8	DoReFa [36]	53.00	55.90	2.90
VGG-16	3	GPFQ (Ours)	69.67	71.59	1.92
		GPFQ (Ours)†	70.24	71.59	1.35
	4	MSE [1]	70.50	71.60	1.10
		OMSE [3]	71.48	73.48	2.00
		GPFQ (Ours)	70.70	71.59	0.89
		GPFQ (Ours)†	70.90	71.59	0.69
	5	GPFQ (Ours)	70.96	71.59	0.63
		GPFQ (Ours)†	71.05	71.59	0.54
	8	Lee et al. [18]	68.05	68.34	0.29
ResNet-18	3	GPFQ (Ours)	66.55	69.76	3.21
		GPFQ (Ours)†	67.63	69.76	2.13
	4	MSE [1]	67.00	69.70	2.70
		OMSE [3]	68.38	69.64	1.26
		S-AdaQuant [14]	69.40	71.97	2.57
		AdaRound [23]	68.71	69.68	0.97
		BRECQ [19]	70.70	71.08	0.38
		GPFQ (Ours)	68.55	69.76	1.21
		GPFQ (Ours)†	68.81	69.76	0.95
	5	RQ [21]	65.10	69.54	4.44
		GPFQ (Ours)	69.27	69.76	0.49
		GPFQ (Ours)†	69.50	69.76	0.26
	6	DFQ [24]	66.30	70.50	4.20
		RQ [21]	68.65	69.54	0.89
ResNet-50	3	GPFQ (Ours)	71.80	76.13	4.33
		GPFQ (Ours)†	72.18	76.13	3.95
	4	MSE [1]	73.80	76.10	2.30
		OMSE [3]	73.39	76.01	2.62
		OCS + Clip [34]	69.30	76.10	6.80
		PWLQ [8]	73.70	76.10	2.40
		AdaRound [23]	75.23	76.07	0.84
		S-AdaQuant [14]	75.10	77.20	2.10
		BRECQ [19]	76.29	77.00	0.71
		GPFQ (Ours)	75.10	76.13	1.03
		GPFQ (Ours)†	75.30	76.13	0.83
	5	OCS + Clip [34]	73.40	76.10	2.70
		GPFQ (Ours)	75.51	76.13	0.62
		GPFQ (Ours)†	75.66	76.13	0.47
	8	IAOI [15]	74.90	76.40	1.50

Further Improvement of GPFQ

In this section, we show that the validation accuracy of the proposed approach can be further improved by incorporating the following modifications used by prior work:



(a) AlexNet with (4.35)

(b) AlexNet with (4.36)

(c) VGG-16 with (4.35)

(d) VGG-16 with (4.36)

(e) ResNet-50 with (4.35)

(f) ResNet-50 with (4.36)

Figure 4.2. (1) Left y-axis: Top-1 (dashed-dotted lines) and Top-5 (dash lines) accuracy for original (in red) and quantized (in blue) models on ImageNet. (2) Right y-axis: The sparsity of quantized models plotted by dotted green lines.

(1) mixing precision for quantization, such as using different bit-widths to quantize fully-connected and convolutional layers respectively [2] or leaving the last fully-connected layer

unquantized [36]; (2) applying bias correction [1, 24] to the last layer, that is, subtracting the average quantization error from the layer's bias term. In Table 4.2, we examine some of these empirical rules by leaving the last layer intact and performing bias correction to remove the noise due to quantization. This variant of GPFQ is highlighted by a † symbol. By using the enhanced GPFQ, the average increment of accuracy exceeds 0.2% for $b = 4, 5$ bits, and is greater than 0.7% for $b = 3$ bits. This demonstrates, empirically, that GPFQ can be easily adapted to incorporate heuristic modifications that improve performance.

Sparse Quantization

For our final experiment, we illustrate the effects of sparsity via the sparse quantization introduced in Section 4.4. Recall that the sparse GPFQ with soft thresholding in (4.35) uses alphabets $\mathscr{A}_K^\delta$ as in (4.4) while the version of hard thresholding, see (4.36), relies on alphabets $\mathscr{A}_K^{\delta,\lambda}$ as in Equation (4.6). In the setting of our experiment, both K and δ are still defined and computed as in Section 4.5.1, where the number of bits $b = 5$ and the corresponding scalar $C > 0$ and batch size $m \in \mathbb{N}$ for each neural network is provided by Table 4.1. Moreover, the *sparsity* of a given neural network is defined as the proportion of zeros in the weights. According to Equation (4.35) and Equation (4.36), in general, the sparsity of DNNs is boosted as λ increases. Hence, we treat $\lambda > 0$ as a variable to control sparsity and explore its impact on validation accuracy of different DNNs. As shown in Figure 4.2, we quantize AlexNet, VGG-16, and ResNet-50 using both (4.35) and (4.36), with $\lambda \in \{0, 0.0025, 0.005, 0.0075, 0.01, 0.0125\}$. Curves for validation accuracy and sparsity are plotted against λ. We note that, for all tested models, sparse GPFQ with hard thresholding, i.e. (4.36), outperforms soft thresholding, achieving significantly higher sparsity and better accuracy. For example, by quantizing AlexNet and VGG-16 with (4.36), one can maintain near-original model accuracy when half the weights are quantized to zero, which implies a remarkable compression rate $\frac{0.5b}{32} = \frac{2.5}{32} \approx 7.8\%$. Similarly, Figure 4.2f and Figure 4.2e show that ResNet-50 can attain 40% sparsity with subtle decrement in accuracy. Additionally, in all cases, one can expect to get higher sparsity by increasing λ while the

validation accuracy tends to drop gracefully. Moreover, in Figure 4.2e, we observe that the sparsity of quantized ResNet50 with $\lambda = 0.0025$ is even lower than the result when thresholding functions are not used, that is, $\lambda = 0$. A possible reason is given as follows. In contrast with $\mathscr{A}_K^{\delta}$, the alphabet $\mathscr{A}_K^{\delta,\lambda}$ has only one element 0 between $-\lambda$ and λ. Thus, to compensate for the lack of small alphabet elements and also reduce the path following error, sparse GPFQ in (4.36) converts more weights to nonzero entries of $\mathscr{A}_K^{\delta,\lambda}$, which in turn dampens the upward trend in sparsity.

4.6 Useful Lemmata

Lemma 4.6.1. *In the context of* (4.10)*, we have* $q_t = \mathscr{Q}\Big(\frac{\langle \widetilde{X}_t^{(i-1)}, u_{t-1}+w_t X_t^{(i-1)}\rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2}\Big)$*. Here, we suppose* $\widetilde{X}_t^{(i-1)} \neq 0$*.*

Proof. According to (4.10), $q_t = \arg\min_{p\in\mathscr{A}} \big\|u_{t-1} + w_t X_t^{(i-1)} - p\widetilde{X}_t^{(i-1)}\big\|_2^2$. Expanding the square and removing the terms irrelevant to p, we obtain

$$\begin{aligned}
q_t &= \arg\min_{p\in\mathscr{A}} \Big(p^2 \|\widetilde{X}_t^{(i-1)}\|_2^2 - 2p\langle \widetilde{X}_t^{(i-1)}, u_{t-1} + w_t X_t^{(i-1)}\rangle \Big) \\
&= \arg\min_{p\in\mathscr{A}} \Big(p^2 - 2p \cdot \frac{\langle \widetilde{X}_t^{(i-1)}, u_{t-1} + w_t X_t^{(i-1)}\rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2} \Big) \\
&= \arg\min_{p\in\mathscr{A}} \Big(p - \frac{\langle \widetilde{X}_t^{(i-1)}, u_{t-1} + w_t X_t^{(i-1)}\rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2} \Big)^2 \\
&= \arg\min_{p\in\mathscr{A}} \Big| p - \frac{\langle \widetilde{X}_t^{(i-1)}, u_{t-1} + w_t X_t^{(i-1)}\rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2} \Big| \\
&= \mathscr{Q}\Big(\frac{\langle \widetilde{X}_t^{(i-1)}, u_{t-1} + w_t X_t^{(i-1)}\rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2} \Big).
\end{aligned}$$

In the last equality, we used the definition of (4.7). □

Lemma 4.6.2. *Suppose* $\widetilde{X}_t^{(i-1)} \neq 0$*. The closed-form expression of* q_t *in* (4.34) *is given by* $q_t = \mathscr{Q} \circ s_\lambda \Big(\frac{\langle \widetilde{X}_t^{(i-1)}, u_{t-1}+w_t X_t^{(i-1)}\rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2} \Big)$*. Here,* $s_\lambda(x) := \operatorname{sign}(x)\max\{|x| - \lambda, 0\}$ *is the soft thresholding function.*

Proof. Expanding the square and removing the terms irrelevant to p, we obtain

$$\begin{aligned} q_t &= \arg\min_{p\in\mathscr{A}}\Big(\frac{p^2}{2}\|\widetilde{X}_t^{(i-1)}\|_2^2 - p\langle\widetilde{X}_t^{(i-1)}, u_{t-1}+w_tX_t^{(i-1)}\rangle + \lambda|p|\|\widetilde{X}_t^{(i-1)}\|_2^2\Big) \\ &= \arg\min_{p\in\mathscr{A}}\Big(\frac{p^2}{2} - p\cdot\frac{\langle\widetilde{X}_t^{(i-1)}, u_{t-1}+w_tX_t^{(i-1)}\rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2} + \lambda|p|\Big) \\ &= \arg\min_{p\in\mathscr{A}}\Big(\frac{p^2}{2} - \alpha_t p + \lambda|p|\Big) \end{aligned} \tag{4.41}$$

where $\alpha_t := \frac{\langle\widetilde{X}_t^{(i-1)}, u_{t-1}+w_tX_t^{(i-1)}\rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2}$. Define $g_t(p) := \frac{1}{2}p^2 - \alpha_t p + \lambda|p|$ for $p\in\mathbb{R}$. By (4.4), we have $q_t = \arg\min_{p\in\mathscr{A}} g_t(p) = \arg\min_{\substack{|k|\le K\\ k\in\mathbb{Z}}} g_t(k\delta)$. Now we analyze two cases $\alpha_t \ge 0$ and $\alpha_t < 0$. The idea is to investigate the behaviour of $g_t(k\delta)$ over $k\in\{-K,...,K\}$.

(I) Assume $\alpha_t \ge 0$. Since $g_t(k\delta) > g_t(0) = 0$ for all $-K \le k \le -1$, then $g_t(k\delta)$ is minimized at some $k \ge 0$. Note that $g_t(p)$ is a convex function passing through the origin. So, for $1 \le k \le K-1$, $g_t(k\delta)$ is the minimum if and only if $g_t(k\delta) \le \min\{g_t((k+1)\delta), g_t((k-1)\delta)\}$.

It is easy to verify that the condition above is equivalent to

$$\Big(k-\frac{1}{2}\Big)\delta + \lambda \le \alpha_t \le \Big(k+\frac{1}{2}\Big)\delta + \lambda. \tag{4.42}$$

It only remains to check $k=0$ and $k=K$. For $k=0$, note that when $\alpha_t \in [0, \delta/2+\lambda]$, we have

$$g_t(\delta) \ge g_t(0) = 0, \tag{4.43}$$

and if $\alpha_t \ge (K-\frac{1}{2})\delta + \lambda$, then

$$g_t(K\delta) \le g_t((K-1)\delta). \tag{4.44}$$

Combining (4.42), (4.43), and (4.44), we conclude that

$$q_t = \arg\min_{\substack{|k|\le K \\ k\in\mathbb{Z}}} g_t(k\delta) = \begin{cases} 0 & \text{if } 0 \le \alpha_t < \frac{\delta}{2} + \lambda, \\ k\delta & \text{if } |\alpha_t - \lambda - k\delta| \le \frac{\delta}{2} \text{ and } 1 \le k \le K-1, \\ K\delta & \text{if } \alpha_t \ge \lambda + \frac{\delta}{2} + (K-1)\delta. \end{cases} \tag{4.45}$$

(II) In the opposite case where $\alpha_t < 0$, it suffices to minimize $g_t(k\delta)$ with $k \le 0$ because $g_t(k\delta) > 0$ for all $k \ge 1$. Again, notice that $g_t(p)$ is a convex function on $[-\infty, 0]$ satisfying $g_t(0) = 0$. Applying a similar argument as in the case $\alpha_t \ge 0$, one can get

$$q_t = \arg\min_{\substack{|k|\le K \\ k\in\mathbb{Z}}} g_t(k\delta) = \begin{cases} 0 & \text{if } -\frac{\delta}{2} - \lambda < \alpha_t < 0, \\ k\delta & \text{if } |\alpha_t + \lambda - k\delta| \le \frac{\delta}{2} \text{ and } -(K-1) \le k \le -1, \\ -K\delta & \text{if } \alpha_t \le -\lambda - \frac{\delta}{2} - (K-1)\delta. \end{cases} \tag{4.46}$$

It follows from (4.45) and (4.46) that $q_t = \mathscr{Q}(s_\lambda(\alpha_t)) = \mathscr{Q} \circ s_\lambda\left(\frac{\langle \widetilde{X}_t^{(i-1)}, u_{t-1} + w_t X_t^{(i-1)} \rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2}\right)$ where $s_\lambda(x) := \operatorname{sign}(x)\max\{|x| - \lambda, 0\}$ is the soft thresholding function. □

Orthogonal Projections. Given a closed subspace $S \subseteq \mathbb{R}^m$, we denote the orthogonal projection onto S by P_S. In particular, if $z \in \mathbb{R}^m$ is a vector, then we use P_z and $P_{z^\perp}$ to represent orthogonal projections onto $\operatorname{span}(z)$ and $\operatorname{span}(z)^\perp$ respectively. Hence, for any $x \in \mathbb{R}^m$, we have

$$P_z(x) = \frac{\langle z, x\rangle z}{\|z\|_2^2}, \quad x = P_z(x) + P_{z^\perp}(x), \quad \text{and} \quad \|x\|_2^2 = \|P_z(x)\|_2^2 + \|P_{z^\perp}(x)\|_2^2. \tag{4.47}$$

Lemma 4.6.3. *Let $\mathscr{A}$ be as in* (4.4) *with step size $\delta > 0$, and largest element q_{max}. Suppose that $w \in \mathbb{R}^{N_0}$ satisfies $\|w\|_\infty \le q_{\max}$, and consider the quantization scheme given by* (4.12)*. Let*

$\theta_t := \angle(X_t, u_{t-1})$ *be the angle between* X_t *and* u_{t-1}*. Then, for* $t = 1, 2, \ldots, N_0$*, we have*

$$\|u_t\|_2^2 - \|u_{t-1}\|_2^2 \leq \begin{cases} \frac{\delta^2}{4}\|X_t\|_2^2 - \|u_{t-1}\|_2^2 \cos^2\theta_t & \text{if } \left|w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\right| \leq q_{max}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.48}$$

Proof. By applying (4.47) and (4.12), we get

$$\begin{aligned} \|P_{X_t}(u_t)\|_2^2 &= \frac{(X_t^\top u_t)^2}{\|X_t\|_2^2} = \frac{(X_t^\top u_{t-1} + (w_t - q_t)\|X_t\|_2^2)^2}{\|X_t\|_2^4}\|X_t\|_2^2 \\ &= \left(w_t + \frac{X_t^\top u_{t-1}}{\|X_t\|_2^2} - q_t\right)^2 \|X_t\|_2^2 = \left(w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t - q_t\right)^2 \|X_t\|_2^2. \end{aligned} \tag{4.49}$$

The last equation holds because $X_t^\top u_{t-1} = \|X_t\|_2\|u_{t-1}\|_2\cos\theta_t$. Note that

$$\left(w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t - q_t\right)^2 - \left(\frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\right)^2 = \underbrace{\left(w_t + \frac{2\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t - q_t\right)}_{\text{(I)}}\underbrace{(w_t - q_t)}_{\text{(II)}},$$

$|w_t| \leq q_{\max}$, and $q_t = \mathscr{Q}\left(w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\right)$. If $\left(w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\right) > q_{\max}$, then $q_t = q_{\max}$ and thus $0 \leq q_t - w_t \leq \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t$. So (I) $\geq w_t + 2(q_t - w_t) - q_t = q_t - w_t \geq 0$ and (II) ≤ 0. Moreover, if $\left(w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\right) < -q_{\max}$, then $q_t = -q_{\max}$ and $\frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t \leq q_t - w_t \leq 0$. Hence, (I) $\leq w_t + 2(q_t - w_t) - q_t = q_t - w_t \leq 0$ and (II) ≥ 0. It follows that

$$\left(w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t - q_t\right)^2 \leq \left(\frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\right)^2 \tag{4.50}$$

when $\left|w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\right| > q_{\max}$. Now, assume that $\left|w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\right| \leq q_{\max}$. In this case, since the argument of $\mathscr{Q}$ lies in the active range of $\mathscr{A}$, we obtain

$$\left(w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t - q_t\right)^2 \leq \frac{\delta^2}{4}. \tag{4.51}$$

Applying (4.50) and (4.51) to (4.49), one can get

$$\|P_{X_t}(u_t)\|_2^2 \leq \begin{cases} \frac{\delta^2}{4}\|X_t\|_2^2 & \text{if } \left| w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t \right| \leq q_{\max}, \\ \|u_{t-1}\|_2^2 \cos^2\theta_t & \text{otherwise.} \end{cases} \tag{4.52}$$

Further, we have

$$P_{X_t^\perp}(u_t) = P_{X_t^\perp}(u_{t-1} + w_t X_t - q_t X_t) = P_{X_t^\perp}(u_{t-1}). \tag{4.53}$$

It follows that

$$\begin{aligned} \|u_t\|_2^2 - \|u_{t-1}\|_2^2 &= \|P_{X_t}(u_t)\|_2^2 + \|P_{X_t^\perp}(u_t)\|_2^2 - \|u_{t-1}\|_2^2 \\ &= \|P_{X_t}(u_t)\|_2^2 + \|P_{X_t^\perp}(u_{t-1})\|_2^2 - \|u_{t-1}\|_2^2 && \text{(by (4.53))} \\ &= \|P_{X_t}(u_t)\|_2^2 - \|P_{X_t}(u_{t-1})\|_2^2 && \text{(using (4.47))} \\ &= \|P_{X_t}(u_t)\|_2^2 - \|u_{t-1}\|_2^2 \cos^2\theta_t. \end{aligned}$$

Substituting $\|P_{X_t}(u_t)\|_2^2$ with its upper bounds in (4.52), we obtain (4.48). □

Lemma 4.6.4. *Suppose that we quantize $w \in \mathbb{R}^{N_0}$ using quantization scheme given by* (4.12). *Additionally, denote the information of the first $t-1$ quantization steps by a σ-algebra $\mathscr{F}_{t-1}$, and let $\beta, \eta > 0$. Then, for $t = 1, 2, \ldots, N_0$, we have*

$$\mathbb{E}(e^{\eta\beta\|u_{t-1}\|_2^2(1-\cos^2\theta_t)} \mid \mathscr{F}_{t-1}) \leq -\mathbb{E}(\cos^2\theta_t \mid \mathscr{F}_{t-1})(e^{\eta\beta\|u_{t-1}\|_2^2} - 1) + e^{\eta\beta\|u_{t-1}\|_2^2}$$

where θ_t is the angle between X_t and u_{t-1}.

Proof. Conditioning on $\mathscr{F}_{t-1}$, the function $f(x) = e^{\eta\beta x\|u_{t-1}\|_2^2}$ is convex. It follows that

$$\begin{aligned}
\mathbb{E}(e^{\eta\beta\|u_{t-1}\|_2^2(1-\cos^2\theta_t)} \mid \mathscr{F}_{t-1}) &= \mathbb{E}(f(\cos^2\theta_t \cdot 0 + (1-\cos^2\theta_t)\cdot 1) \mid \mathscr{F}_{t-1}) \\
&\leq \mathbb{E}(\cos^2\theta_t + (1-\cos^2\theta_t)e^{\eta\beta\|u_{t-1}\|_2^2} \mid \mathscr{F}_{t-1}) \\
&\leq \mathbb{E}(\cos^2\theta_t \mid \mathscr{F}_{t-1}) + (1-\mathbb{E}(\cos^2\theta_t \mid \mathscr{F}_{t-1}))e^{\eta\beta\|u_{t-1}\|_2^2} \\
&= -\mathbb{E}(\cos^2\theta_t \mid \mathscr{F}_{t-1})(e^{\eta\beta\|u_{t-1}\|_2^2} - 1) + e^{\eta\beta\|u_{t-1}\|_2^2}.
\end{aligned}$$

□

4.7 Fusing Convolution and Batch Normalization Layers

For many neural networks, e.g. MobileNets and ResNets, a convolutional layer is usually followed by a batch normalization (BN) layer to normalize the output. Here, we show how our quantization approach admits a simple modification that takes into account such BN layers. Specifically, denote the convolution operator by * and suppose that a convolutional layer

$$f_{\text{conv}}(x) := w_{\text{conv}} * x + b_{\text{conv}} \tag{4.54}$$

is followed by a BN layer given by

$$f_{\text{bn}}(x) := \frac{x-\hat{\mu}}{\sqrt{\hat{\sigma}^2+\varepsilon}} \cdot w_{\text{bn}} + b_{\text{bn}}. \tag{4.55}$$

Here, w_{conv}, w_{bn}, b_{conv}, and b_{bn} are learned parameters and $\hat{\mu}$, $\hat{\sigma}$ are the running mean and standard-deviation respectively while $\varepsilon > 0$ is to keep the denominator bounded away from 0. Note that the parameters in both Equation (4.54) and Equation (4.55) are calculated per-channel over the mini-batches during training, but fixed thereafter.

Thus, to quantize the convolutional and subsequent BN layers simultaneously, we first

Table 4.3. Top-1 accuracy drop for ResNet-18 and ResNet-50.

Model	b	m	Unfused		Fused	
			C	Acc Drop (%)	C	Acc Drop (%)
ResNet-18	4	2048	1.16	1.63	1.29	1.72
	4	4096		1.21		1.18
	5	2048		0.71		0.72
	5	4096		0.49		0.51
ResNet-50	5	512	1.81	0.97	1.82	1.03
	5	1024		0.90		0.81
	5	2048		0.62		0.64

observe that we can write

$$f_{\text{bn}} \circ f_{\text{conv}}(x) = w_{\text{new}} * x + b_{\text{new}} \tag{4.56}$$

with

$$w_{\text{new}} := \frac{w_{\text{conv}} w_{\text{bn}}}{\sqrt{\hat{\sigma}^2 + \varepsilon}}, \quad b_{\text{new}} := \frac{(b_{\text{conv}} - \hat{\mu}) w_{\text{bn}}}{\sqrt{\hat{\sigma}^2 + \varepsilon}} + b_{\text{bn}}.$$

As a result, to quantize the convolutional and subsequent BN layer simulatenously, we can simply quantize the parameters $w_{\text{new}}, b_{\text{new}}$ in (4.56) using our methods. Although BN layers are not quantized in our experiments in Section 4.5, we will show here that the proposed algorithm GPFQ is robust to neural network fusion as described above. In Table 4.3, we compare the Top-1 quantization accuracy between fused ResNets and unfused ResNets when quantized using our methods with different bits and batch sizes. Note that the scalar C for unfused networks remains the same as in Table 4.1 while C for fused networks is selected using the procedure after Equation (4.40). We observe that the performance of GPFQ for fused ResNet-18 and ResNet-50 is quite similar to that for unfused networks.

4.8 Quantizing Large Weights

In this section, we demonstrate that the proposed quantization algorithm (4.12) is still effective for weights with magnitudes that exceed the largest element, $q_{\max} = K\delta$, in the alphabet set $\mathscr{A}$.

Specifically, we prove Theorem 4.8.2, bounding the expected error when $n := n(\delta)$ entries of w are greater than $K\delta$. In turn, Theorem 4.8.2 suggests that in some cases, choosing δ such that $n(\delta) > 0$ may be advantageous, a finding that is consistent with our experiments in Section 4.5. We begin with the following lemma needed to prove Theorem 4.8.2.

Lemma 4.8.1. *Let $\mathscr{A}$ be as in* (4.4) *with step size $\delta > 0$, and largest element q_{max}. Suppose that $w \in \mathbb{R}^{N_0}$ satisfies $\|w\|_\infty \leq k q_{\max}$ for some $k > 1$, and consider the quantization scheme given by* (4.12). *Let $\theta_t := \angle(X_t, u_{t-1})$ be the angle between X_t and u_{t-1}. Then*

$$\|u_t\|_2^2 \leq \begin{cases} \frac{\delta^2}{4}\|X_t\|_2^2 + \|u_{t-1}\|_2^2(1-\cos^2\theta_t) & \text{if } \left|w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\right| \leq q_{\max}, \\ \|u_{t-1}\|_2^2 & \text{if } \left|w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\right| > q_{\max} \text{ and } |w_t| \leq q_{\max}, \\ (\|u_{t-1}\|_2 + (k-1)q_{\max}\|X_t\|_2)^2 & \text{if } \left|w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\right| > q_{\max} \text{ and } |w_t| > q_{\max} \end{cases} \tag{4.57}$$

holds for $t = 1, 2, \ldots, N_0$.

Proof. The first two cases in (4.57) are covered by Lemma 4.6.3. So it remains to consider the case where $\left|w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\right| > q_{\max}$ and $|w_t| > q_{\max}$. As in the proof of Lemma 4.6.3, we have

$$\|u_t\|_2^2 = (v_t - q_t)^2\|X_t\|_2^2 + (1-\cos^2\theta_t)\|u_{t-1}\|_2^2$$

where $v_t := w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t$. Since $q_t = \mathscr{Q}(v_t)$ and $|v_t| > q_{\max}$, we get $q_t = \operatorname{sign}(v_t)q_{\max}$. It follows that

$$\begin{aligned} \|u_t\|_2^2 &= (v_t - \operatorname{sign}(v_t)q_{\max})^2\|X_t\|_2^2 + (1-\cos^2\theta_t)\|u_{t-1}\|_2^2 \\ &= (|v_t| - q_{\max})^2\|X_t\|_2^2 + (1-\cos^2\theta_t)\|u_{t-1}\|_2^2. \end{aligned} \tag{4.58}$$

By symmetry, we can assume without loss of generality that $v_t > q_{\max}$. In this case, since

$|w_t| \leq \|w\|_\infty \leq k q_{\max}$,

$$|v_t| - q_{\max} = v_t - q_{\max} = w_t - q_{\max} + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t \leq (k-1)q_{\max} + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t.$$

Then (4.58) becomes

$$\begin{aligned}
\|u_t\|_2^2 &\leq \Big((k-1)q_{\max} + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\Big)^2 \|X_t\|_2^2 + (1-\cos^2\theta_t)\|u_{t-1}\|_2^2 \\
&= (k-1)^2 q_{\max}^2 \|X_t\|_2^2 + \|u_{t-1}\|_2^2 + 2(k-1)q_{\max}\langle X_t, u_{t-1}\rangle \\
&= \|(k-1)q_{\max}X_t + u_{t-1}\|_2^2 \\
&\leq (\|u_{t-1}\|_2 + (k-1)q_{\max}\|X_t\|_2)^2.
\end{aligned}$$

This completes the proof. □

We are now ready to bound the expected quantization error in the case when some weights have magnitude greater than $q_{\max}$.

Theorem 4.8.2. *Suppose that the columns X_t of $X \in \mathbb{R}^{m\times N_0}$ are drawn independently from a probability distribution for which there exists $s \in (0,1)$ and $r > 0$ such that $\|X_t\|_2 \leq r$ almost surely, and such that for all unit vector $u \in \mathbb{S}^{m-1}$ we have*

$$\mathbb{E}\Big(\frac{\langle X_t, u\rangle^2}{\|X_t\|_2^2}\,\Big|\,\mathscr{H}_t\Big)\mathrm{P}(\mathscr{H}_t) \geq s^2. \tag{4.59}$$

Here, $\mathscr{H}_t$ represents the event $\{|w_t + \frac{\langle X_t, u_{t-1}\rangle}{\|X_t\|_2^2}| \leq q_{max}\}$. Let $\mathscr{A}$ be the alphabet in (4.4) *with step size $\delta > 0$, and the largest element $q_{\max}$. Let $w \in \mathbb{R}^{N_0}$ be the weights associated with a neuron such that $\|w\|_\infty \leq k q_{\max}$ for some $k > 1$. Let $n = |\{t : |w_t| > q_{\max}\}|$ be the number of weights with magnitude greater than $q_{\max}$. Quantizing w using* (4.12)*, we have*

$$\mathbb{E}\|Xw - Xq\|_2^2 \leq \Big(nr(k-1)q_{\max} + \frac{1}{2}nr\delta + \frac{\delta r}{2s}\Big)^2. \tag{4.60}$$

Proof. Let $\mathscr{E}_t$ represent the event $\{|w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t| \le q_{\max}\}$ where θ_t is the angle between X_t and u_{t-1}. Denote the information of the first $t-1$ quantization steps by $\mathscr{F}_{t-1}$. Additionally, we define

$$p_t := \mathrm{P}(\mathscr{E}_t \mid \mathscr{F}_{t-1}) \quad \text{and} \quad s_t^2 := \mathbb{E}\Big(\frac{\langle X_t, u_{t-1}\rangle^2}{\|X_t\|_2^2\|u_{t-1}\|_2^2} \;\Big|\; \mathscr{F}_{t-1}, \mathscr{E}_t\Big).$$

By (4.59), we have

$$p_t s_t^2 \ge s^2. \tag{4.61}$$

Since $\|X_t\|_2 \le r$ almost surely, by Lemma 4.8.1, we obtain

$$\mathbb{E}(\|u_t\|_2^2 \mid \mathscr{F}_{t-1}, \mathscr{E}_t) \le \frac{1}{4}\delta^2 r^2 + (1-s_t^2)\|u_{t-1}\|_2^2 \tag{4.62}$$

and

$$\mathbb{E}(\|u_t\|_2^2 \mid \mathscr{F}_{t-1}, \mathscr{E}_t^c) \le \begin{cases} \|u_{t-1}\|_2^2 & \text{if } |w_t| \le q_{\max}, \\ (\|u_{t-1}\|_2 + (k-1)rq_{\max})^2 & \text{if } |w_t| > q_{\max}. \end{cases} \tag{4.63}$$

Moreover, we have

$$\begin{aligned} \mathbb{E}(\|u_t\|_2^2 \mid \mathscr{F}_{t-1}) &= \mathbb{E}(\|u_t\|_2^2 \mid \mathscr{F}_{t-1}, \mathscr{E}_t)\mathrm{P}(\mathscr{E}_t \mid \mathscr{F}_{t-1}) + \mathbb{E}(\|u_t\|_2^2 \mid \mathscr{F}_{t-1}, \mathscr{E}_t^c)\mathrm{P}(\mathscr{E}_t^c \mid \mathscr{F}_{t-1}) \\ &= p_t\mathbb{E}(\|u_t\|_2^2 \mid \mathscr{F}_{t-1}, \mathscr{E}_t) + (1-p_t)\mathbb{E}(\|u_t\|_2^2 \mid \mathscr{F}_{t-1}, \mathscr{E}_t^c). \end{aligned} \tag{4.64}$$

If $|w_t| \le q_{\max}$, then using (4.64), (4.63), and (4.62), we obtain

$$\begin{aligned} \mathbb{E}(\|u_t\|_2^2 \mid \mathscr{F}_{t-1}) &\le \frac{1}{4}p_t\delta^2 r^2 + p_t(1-s_t^2)\|u_{t-1}\|_2^2 + (1-p_t)\|u_{t-1}\|_2^2 \\ &\le \frac{1}{4}\delta^2 r^2 + (1-s^2)\|u_{t-1}\|_2^2. \end{aligned}$$

In the last step, we used (4.61) and $p_t \in [0,1]$. Similarly, if $|w_t| > q_{\max}$, then

$$\begin{aligned}
&\mathbb{E}(\|u_t\|_2^2 \mid \mathscr{F}_{t-1}) \leq \frac{1}{4} p_t \delta^2 r^2 + p_t(1-s_t^2)\|u_{t-1}\|_2^2 + (1-p_t)(\|u_{t-1}\|_2 + (k-1)rq_{\max})^2 \\
&= (1 - p_t s_t^2)\|u_{t-1}\|_2^2 + 2(k-1)rq_{\max}(1-p_t)\|u_{t-1}\|_2 + \frac{1}{4} p_t \delta^2 r^2 + (1-p_t)(k-1)^2 r^2 q_{\max}^2 \\
&\leq \|u_{t-1}\|_2^2 + 2(k-1)rq_{\max}\|u_{t-1}\|_2 + \frac{1}{4}\delta^2 r^2 + (k-1)^2 r^2 q_{\max}^2.
\end{aligned}$$

Let $a := 1 - s^2$, $b := \frac{1}{4}\delta^2 r^2$, and $c := (k-1)rq_{\max} + \frac{1}{2}\delta r$. It follows that

$$\mathbb{E}\|u_t\|_2^2 = \mathbb{E}(\mathbb{E}(\|u_t\|_2^2 \mid \mathscr{F}_{t-1})) \leq \begin{cases} a\mathbb{E}\|u_{t-1}\|_2^2 + b & \text{if } |w_t| \leq q_{\max}, \\ \mathbb{E}\|u_{t-1}\|_2^2 + 2c\mathbb{E}\|u_{t-1}\|_2 + c^2 & \text{if } |w_t| > q_{\max}. \end{cases} \tag{4.65}$$

Define the indices $t_0 := 0 < t_1 < \ldots < t_n < t_{n+1} := N_0 + 1$ where $|w_{t_j}| > q_{\max}$ and let $m_j := t_j - t_{j-1} - 1$ for $1 \leq j \leq n$. Applying the first case in (4.65) recursively, one obtain

$$\mathbb{E}\|u_{t_1-1}\|_2^2 \leq a^{m_1}\mathbb{E}\|u_0\|_2^2 + b(1 + a + \ldots + a^{m_1-1}) = b(1 + a + \ldots + a^{m_1-1}). \tag{4.66}$$

In the last equation, we used the fact $u_0 = 0$. Next, the second case in (4.65) can be used to bound $\mathbb{E}\|u_{t_1}\|_2^2$. Specifically, we have

$$\begin{aligned}
\mathbb{E}\|u_{t_1}\|_2^2 &\leq \mathbb{E}\|u_{t_1-1}\|_2^2 + 2c\mathbb{E}\|u_{t_1-1}\|_2 + c^2 && \text{(using (4.65))} \\
&\leq \mathbb{E}\|u_{t_1-1}\|_2^2 + 2c(\mathbb{E}\|u_{t_1-1}\|_2^2)^{\frac{1}{2}} + c^2 && \text{(by Jensen's inequality)} \\
&= ((\mathbb{E}\|u_{t_1-1}\|_2^2)^{\frac{1}{2}} + c)^2 \\
&\leq \left(c + \sqrt{b(1 + a + \ldots + a^{m_1-1})}\right)^2 && \text{(using (4.66))}.
\end{aligned} \tag{4.67}$$

Since $|w_t| \leq q_{\max}$ for $t_1 < t < t_2$, using (4.65), we can derive

$$\begin{aligned}
\mathbb{E}\|u_{t_2-1}\|_2^2 &\leq a^{m_2}\mathbb{E}\|u_{t_1}\|_2^2 + b(1+a+\ldots+a^{m_2-1}) \\
&\leq a^{m_2}\left(c+\sqrt{b\cdot\frac{1-a^{m_1}}{1-a}}\right)^2 + b\cdot\frac{1-a^{m_2}}{1-a} && \text{(using (4.67))} \\
&= a^{m_2}c^2 + b\cdot\frac{1-a^{m_1+m_2}}{1-a} + 2a^{m_2}c\sqrt{\frac{b(1-a^{m_1})}{1-a}} \\
&\leq c^2 + b\cdot\frac{1-a^{m_1+m_2}}{1-a} + 2c\sqrt{\frac{b(1-a^{m_1+m_2})}{1-a}} && \text{(since } 0<a<1\text{)} \\
&\leq \left(c+\sqrt{\frac{b(1-a^{m_1+m_2})}{1-a}}\right)^2. && (4.68)
\end{aligned}$$

Hence, we obtain $\mathbb{E}\|u_{t_2-1}\|_2^2 \leq \left(c+\sqrt{\frac{b}{1-a}}\right)^2$. Proceeding in the same way for the remaining indices t_i up to $t_{n+1}-1 = N_0$, we obtain

$$\mathbb{E}\|u_{N_0}\|_2^2 \leq \left(nc+\sqrt{\frac{b}{1-a}}\right)^2 = \left(nr(k-1)q_{\max}+\frac{1}{2}nr\delta+\frac{\delta r}{2s}\right)^2. \tag{4.69}$$

Since $u_{N_0} = Xw - Xq$, we have $\mathbb{E}\|Xw-Xq\|_2^2 \leq (nr(k-1)q_{\max}+\frac{1}{2}nr\delta+\frac{\delta r}{2s})^2$. □

Our numerical experiments in Section 4.5 demonstrated that choosing our alphabet with $q_{\max} < \|w\|_\infty$ can yield better results than if we strictly conformed to choosing $\mathscr{A}$ with $q_{\max} \geq \|w\|_\infty$. Let us now see how Theorem 4.8.2 can help explain these experimental results. First, recall from (4.4) that $q_{\max} = K\delta = 2^{b-1}\delta$ where b is the number of bits, and observe that the condition $\|w\|_\infty \leq kq_{\max}$ in Theorem 4.8.2 implies that we can set $k = \|w\|_\infty/q_{\max}$. Thus (4.60), coupled with Jensen's inequality, yields

$$\mathbb{E}\|Xw-Xq\|_2 \leq nr(\|w\|_\infty - q_{\max}+\frac{1}{2}\delta)+\frac{\delta r}{2s} = nr(\|w\|_\infty-(2^{b-1}-2^{-1})\delta)+\frac{\delta r}{2s}. \tag{4.70}$$

Now, note that s, r are fixed parameters that only depend on the input data distribution so for a

fixed b, $n = n(\delta) = |\{t : |w_t| > 2^{b-1}\delta\}|$ is a decreasing function of δ. In other words, the right hand side of (4.70) is the sum of an increasing function of δ and a decreasing function of δ. This means that there exists an optimal value of δ^* that minimizes the bound. In particular, it may not always be optimal to choose a large δ such that $\|w\|_\infty = 2^{b-1}\delta$. This gives a theoretical justification for why the simple grid search we used in Section 4.5 yielded better results.

4.9 Theoretical Analysis for Gaussian Clusters

In this section, we will prove Theorem 4.3.4, which we first restate here for convenience. Theorem 4.3.4: Let $X \in \mathbb{R}^{m\times N_0}$ be as in (4.30) and let $\mathscr{A}$ be as in (4.3), with step size $\delta > 0$. Let $p \in \mathbb{N}$, $J = 1 + (d\sigma^2)^{-1}\max_{1\le t\le N_0}\sum_{i=1}^d (z_t^{(i)})^2$, and $w \in \mathbb{R}^{N_0}$ be the weights associated with a neuron. Quantizing w using (4.12), we have

$$\mathrm{P}\Big(\|Xw - Xq\|_2^2 \ge 4pm^2J^2\delta^2\sigma^2\log N_0\Big) \le \frac{7\sqrt{mJ}}{N_0^p}, \quad \text{and}$$

$$\mathrm{P}\Big(\max_{1\le t\le N_0}\|u_t\|_2^2 \ge 4pm^2J^2\delta^2\sigma^2\log N_0\Big) \le \frac{7\sqrt{mJ}}{N_0^{p-1}}.$$

If the activation function φ is ξ-Lipschitz continuous, then

$$\mathrm{P}\Big(\|\varphi(Xw) - \varphi(Xq)\|_2^2 \ge 4pm^2J^2\xi^2\delta^2\sigma^2\log N_0\Big) \le \frac{7\sqrt{mJ}}{N_0^p}.$$

4.9.1 Proof of Theorem 4.3.4

Due to $\|X_t\|_2^2 = \sum_{i=1}^d \|Y_t^{(i)}\|_2^2$,

$$\mathbb{E}\|X_t\|_2^2 = \sum_{i=1}^d \mathbb{E}\|Y_t^{(i)}\|_2^2 = \sum_{i=1}^d (n\sigma^2 + n(z_t^{(i)})^2) = m\sigma^2 + n\sum_{i=1}^d (z_t^{(i)})^2 \tag{4.71}$$

Additionally, given a unit vector $u = (u^{(1)}, u^{(2)}, \ldots, u^{(d)}) \in \mathbb{R}^m$ with $u^{(i)} \in \mathbb{R}^n$, we have $\langle X_t, u\rangle = \sum_{i=1}^d \langle Y_t^{(i)}, u^{(i)}\rangle \sim \mathscr{N}\Big(\sum_{i=1}^d z_t^{(i)} u^{(i)\top}\mathbb{1}_n, \sigma^2\Big)$. In fact, once we get the lower bound of $\mathbb{E}\frac{\langle X_t,u\rangle^2}{\|X_t\|_2^2}$

as in (4.13), the quantization error for unbounded data (4.30) can be derived similarly to the proof of Theorem 4.3.1, albeit using different techniques. It follows from the Cauchy-Schwarz inequality that

$$\mathbb{E}\frac{\langle X_t,u\rangle^2}{\|X_t\|_2^2} \geq \frac{(\mathbb{E}|\langle X_t,u\rangle|)^2}{\mathbb{E}\|X_t\|_2^2}. \tag{4.72}$$

$\mathbb{E}\|X_t\|_2^2$ is given by (4.71) while $\mathbb{E}|\langle X_t,u\rangle|$ can be evaluated by the following results.

Lemma 4.9.1. *Let $Z \sim \mathcal{N}(\mu,\sigma^2)$ be a normally distributed random variable. Then*

$$\mathbb{E}|Z| \geq \sigma\sqrt{\frac{2}{\pi}}\Big(1-\frac{4}{27\pi}\Big). \tag{4.73}$$

Proof. Let $\Psi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-t^2/2}\,dt$ be the normal cumulative distribution function. Due to $Z \sim \mathcal{N}(\mu,\sigma^2)$, the folded normal distribution $|Z|$ has mean $\mathbb{E}|Z| = \sigma\sqrt{\frac{2}{\pi}}e^{-\mu^2/2\sigma^2} + \mu(1-2\Psi(-\frac{\mu}{\sigma}))$. A well-known result [9, 29] that can be used to bound $\Psi(x)$ is

$$\int_x^{\infty} e^{-t^2/2}\,dt \leq \min\Big(\sqrt{\frac{\pi}{2}},\frac{1}{x}\Big)e^{-x^2/2}, \quad \text{for} x>0. \tag{4.74}$$

Additionally, in order to evaluate $\mathbb{E}|Z|$, it suffices to analyze the case $\mu \geq 0$ because one can replace Z by $-Z$ without changing $|Z|$ when $\mu < 0$. So we suppose $\mu \geq 0$.

By (4.74), we obtain

$$\begin{aligned}\mathbb{E}|Z| &= \sigma\sqrt{\frac{2}{\pi}}e^{-\mu^2/2\sigma^2} + \mu - 2\mu\Psi(-\mu/\sigma) = \sigma\sqrt{\frac{2}{\pi}}e^{-\mu^2/2\sigma^2} + \mu - \mu\sqrt{\frac{2}{\pi}}\int_{\mu/\sigma}^{\infty} e^{-t^2/2}\,dt \\ &\geq \sigma\sqrt{\frac{2}{\pi}}e^{-\mu^2/2\sigma^2} + \mu - \min\Big(\mu,\sigma\sqrt{\frac{2}{\pi}}\Big)e^{-\mu^2/2\sigma^2}.\end{aligned}$$

If $\mu \geq \sigma\sqrt{\frac{2}{\pi}}$, then one can easily get $\mathbb{E}|Z| \geq \mu \geq \sigma\sqrt{\frac{2}{\pi}}$. Further, if $0 \leq \mu < \sigma\sqrt{\frac{2}{\pi}}$, then $\mathbb{E}|Z| \geq (\sigma\sqrt{2/\pi}-\mu)e^{-\mu^2/2\sigma^2} + \mu$. Due to $e^x \geq 1+x$ for all $x \in \mathbb{R}$, one can get

$$\mathbb{E}|Z| \geq (\sigma\sqrt{2/\pi}-\mu)(1-\mu^2/2\sigma^2) + \mu = \frac{1}{2\sigma^2}\mu^3 - \frac{1}{\sigma\sqrt{2\pi}}\mu^2 + \sigma\sqrt{\frac{2}{\pi}} \geq \sigma\sqrt{\frac{2}{\pi}}\Big(1-\frac{4}{27\pi}\Big).$$

In the last inequality, we optimized in $\mu \in (0, \sigma\sqrt{2/\pi})$ and thus chose $\mu = \frac{2}{3} \cdot \sigma\sqrt{\frac{2}{\pi}}$. □

Lemma 4.9.2. *Let clustered data $X = [X_1, X_2, \ldots, X_{N_0}] \in \mathbb{R}^{m \times N_0}$ be defined as in* (4.30) *and $u \in \mathbb{R}^m$ be a unit vector. Then, for $1 \leq t \leq N_0$, we have*

$$\mathbb{E}\frac{\langle X_t, u\rangle^2}{\|X_t\|_2^2} \geq \frac{5}{9} \cdot \frac{\sigma^2}{m\sigma^2 + n\sum_{i=1}^d (z_t^{(i)})^2}.$$

Proof. Since $\langle X_t, u\rangle$ is normally distributed with variance σ^2, (4.73) implies

$$\mathbb{E}|\langle X_t, u\rangle| \geq \sigma\sqrt{\frac{2}{\pi}}\Big(1 - \frac{4}{27\pi}\Big).$$

Plugging the inequality above and (4.71) into (4.72), we obtain

$$\mathbb{E}\frac{\langle X_t, u\rangle^2}{\|X_t\|_2^2} \geq \frac{(\mathbb{E}|\langle X_t, u\rangle|)^2}{\mathbb{E}\|X_t\|_2^2} \geq \frac{2(1-\frac{4}{27\pi})^2}{\pi} \cdot \frac{\sigma^2}{m\sigma^2 + n\sum_{i=1}^d (z_t^{(i)})^2} \geq \frac{5}{9} \cdot \frac{\sigma^2}{m\sigma^2 + n\sum_{i=1}^d (z_t^{(i)})^2}.$$

□

Now we are ready to prove Theorem 4.3.4.

Proof. Let $\alpha > 0$ and $\eta > 0$. By using exactly the same argument as in (4.17), at the t-th step of (4.12), we have

$$\mathrm{P}(\|u_t\|_2^2 \geq \alpha) \leq e^{-\eta\alpha}\mathbb{E}e^{\eta\|u_t\|_2^2}. \tag{4.75}$$

Moreover, applying Lemma 4.6.3 with $q_{\max} = \infty$, we have

$$\|u_t\|_2^2 \leq \frac{\delta^2}{4}\|X_t\|_2^2 + (1 - \cos^2\theta_t)\|u_{t-1}\|_2^2.$$

It follows that

$$\mathbb{E}e^{\eta\|u_t\|_2^2} \leq \mathbb{E}(e^{\frac{\eta\delta^2}{4}\|X_t\|_2^2} e^{\eta\|u_{t-1}\|_2^2(1-\cos^2\theta_t)}). \tag{4.76}$$

Until now our analysis here has been quite similar to what we did for bounded input data in

Theorem 4.3.1. Nevertheless, unlike Theorem 4.3.1, we will control the moment generating function of $\|X_t\|_2^2$ because $\|X_t\|_2^2$ is unbounded. Specifically, applying the Cauchy-Schwarz inequality and Lemma 4.6.4 with $\beta = 2$, we obtain

$$\begin{aligned}\mathbb{E}(e^{\frac{\eta\delta^2}{4}\|X_t\|_2^2} e^{\eta\|u_{t-1}\|_2^2(1-\cos^2\theta_t)} \mid \mathscr{F}_{t-1}) &\le \big(\mathbb{E}e^{\frac{\eta\delta^2}{2}\|X_t\|_2^2}\big)^{\frac{1}{2}} \big(\mathbb{E}(e^{2\eta\|u_{t-1}\|_2^2(1-\cos^2\theta_t)} \mid \mathscr{F}_{t-1})\big)^{\frac{1}{2}} \\ &\le \big(\mathbb{E}e^{\frac{\eta\delta^2}{2}\|X_t\|_2^2}\big)^{\frac{1}{2}} \big(-\mathbb{E}(\cos^2\theta_t \mid \mathscr{F}_{t-1})(e^{2\eta\|u_{t-1}\|_2^2} - 1) + e^{2\eta\|u_{t-1}\|_2^2}\big)^{\frac{1}{2}} \qquad (4.77)\end{aligned}$$

In the first step, we also used the fact that X_t is independent of $\mathscr{F}_{t-1}$. By Lemma 4.9.2, we have

$$\mathbb{E}(\cos^2\theta_t \mid \mathscr{F}_{t-1}) = \mathbb{E}\Big(\frac{\langle X_t, u_{t-1}\rangle^2}{\|X_t\|_2^2\|u_{t-1}\|_2^2} \Big| \mathscr{F}_{t-1}\Big) \ge \frac{5}{9mJ} =: s^2.$$

Plugging the inequality above into (4.77), we get

$$\begin{aligned}\mathbb{E}(e^{\frac{\eta\delta^2}{4}\|X_t\|_2^2} e^{\eta\|u_{t-1}\|_2^2(1-\cos^2\theta_t)} \mid \mathscr{F}_{t-1}) &\le \big(\mathbb{E}e^{\frac{\eta\delta^2}{2}\|X_t\|_2^2}\big)^{\frac{1}{2}} \Big(-s^2(e^{2\eta\|u_{t-1}\|_2^2} - 1) + e^{2\eta\|u_{t-1}\|_2^2}\Big)^{\frac{1}{2}} \\ &= \big(\mathbb{E}e^{\frac{\eta\delta^2}{2}\|X_t\|_2^2}\big)^{\frac{1}{2}} \Big(e^{2\eta\|u_{t-1}\|_2^2}(1-s^2) + s^2\Big)^{\frac{1}{2}} \\ &\le \big(\mathbb{E}e^{\frac{\eta\delta^2}{2}\|X_t\|_2^2}\big)^{\frac{1}{2}} (e^{\eta\|u_{t-1}\|_2^2}(1-s^2)^{\frac{1}{2}} + s) \\ &\le \big(\mathbb{E}e^{\frac{\eta\delta^2}{2}\|X_t\|_2^2}\big)^{\frac{1}{2}} (e^{\eta\|u_{t-1}\|_2^2}(1 - \frac{1}{2}s^2) + s) \qquad (4.78)\end{aligned}$$

where the last two inequalities hold due to $(x^2+y^2)^{\frac{1}{2}} \le |x| + |y|$ for all $x, y \in \mathbb{R}$, and $(1-x)^{\frac{1}{2}} \le 1 - \frac{1}{2}x$ whenever $x \le 1$.

Now we evaluate $\mathbb{E}e^{\frac{\eta\delta^2}{2}\|X_t\|_2^2}$ and note that

$$\mathbb{E}e^{\frac{\eta\delta^2}{2}\|X_t\|_2^2} = \mathbb{E}\exp\Big(\frac{\eta\delta^2}{2}\sum_{i=1}^{d}\|Y_t^{(i)}\|_2^2\Big) = \prod_{i=1}^{d}\mathbb{E}\exp\Big(\frac{\eta\delta^2}{2}\|Y_t^{(i)}\|_2^2\Big). \qquad (4.79)$$

Since $Y_t^{(i)} \sim \mathcal{N}(z_t^{(i)} \mathbb{1}_n, \sigma^2 I_n)$, we have

$$\begin{aligned}
\mathbb{E}\exp\Big(\frac{\eta\delta^2}{2}\|Y_t^{(i)}\|_2^2\Big) &= \Big[\frac{1}{\sigma\sqrt{2\pi}}\int_{\mathbb{R}}\exp\Big(-\frac{(x-z_t^{(i)})^2}{2\sigma^2}+\frac{\eta\delta^2 x^2}{2}\Big)dx\Big]^n \\
&= \Big\{\frac{1}{\sigma\sqrt{2\pi}}\cdot\exp\Big(\frac{\eta\delta^2(z_t^{(i)})^2}{2-2\eta\delta^2\sigma^2}\Big)\int_{\mathbb{R}}\exp\Big[-\frac{1-\eta\delta^2\sigma^2}{2\sigma^2}\Big(x-\frac{z_t^{(i)}}{1-\eta\delta^2\sigma^2}\Big)^2\Big]dx\Big\}^n \\
&= \Big[(1-\eta\delta^2\sigma^2)^{-\frac{1}{2}}\exp\Big(\frac{\eta\delta^2(z_t^{(i)})^2}{2-2\eta\delta^2\sigma^2}\Big)\Big]^n
\end{aligned}$$

where the last equality holds if $\eta\delta^2\sigma^2 < 1$ and we use the integral of the normal density function:

$$\Big(\frac{1-\eta\delta^2\sigma^2}{2\pi\sigma^2}\Big)^{\frac{1}{2}}\int_{\mathbb{R}}\exp\Big[-\frac{1-\eta\delta^2\sigma^2}{2\sigma^2}\Big(x-\frac{z_t^{(i)}}{1-\eta\delta^2\sigma^2}\Big)^2\Big]dx = 1.$$

Notice that $\frac{1}{1-x} \le 1+2x$ for $x \in [0,\frac{1}{2}]$ and $1+x \le e^x$ for all $x \in \mathbb{R}$. Now, we suppose $\eta\delta^2\sigma^2 \le \frac{1}{2}$ and thus $(1-\eta\delta^2\sigma^2)^{-\frac{1}{2}} = \Big(\frac{1}{1-\eta\delta^2\sigma^2}\Big)^{\frac{1}{2}} \le (1+2\eta\delta^2\sigma^2)^{\frac{1}{2}} \le e^{\eta\delta^2\sigma^2}$. It follows that

$$\begin{aligned}
\mathbb{E}\exp\Big(\frac{\eta\delta^2}{2}\|Y_t^{(i)}\|_2^2\Big) &\le \Big[\exp\Big(\eta\delta^2\sigma^2+\frac{\eta\delta^2(z_t^{(i)})^2}{2-2\eta\delta^2\sigma^2}\Big)\Big]^n \le \Big[\exp\Big(\eta\delta^2\sigma^2+\eta\delta^2(z_t^{(i)})^2\Big)\Big]^n \\
&\le \exp\Big(n\eta\delta^2\sigma^2\Big(1+\frac{(z_t^{(i)})^2}{\sigma^2}\Big)\Big) \qquad (4.80)
\end{aligned}$$

Substituting (4.80) into (4.79), we get

$$\mathbb{E}e^{\frac{\eta\delta^2}{2}\|X_t\|_2^2} \le e^{ndJ\eta\delta^2\sigma^2} = e^{mJ\eta\delta^2\sigma^2}. \qquad (4.81)$$

Combining (4.78) and (4.81), if $\eta\delta^2\sigma^2 \le \frac{1}{2}$, then

$$\begin{aligned}
\mathbb{E}(e^{\frac{\eta\delta^2}{4}\|X_t\|_2^2} e^{\eta\|u_{t-1}\|_2^2(1-\cos^2\theta_t)}) &= \mathbb{E}\Big(\mathbb{E}(e^{\frac{\eta\delta^2}{4}\|X_t\|_2^2} e^{\eta\|u_{t-1}\|_2^2(1-\cos^2\theta_t)} \mid \mathscr{F}_{t-1})\Big) \\
&\le \mathbb{E}\Big(e^{\frac{1}{2}mJ\eta\delta^2\sigma^2}(e^{\eta\|u_{t-1}\|_2^2}(1-\frac{1}{2}s^2)+s)\Big) \\
&= e^{\frac{1}{2}mJ\eta\delta^2\sigma^2}(1-\frac{1}{2}s^2)\mathbb{E}e^{\eta\|u_{t-1}\|_2^2} + se^{\frac{1}{2}mJ\eta\delta^2\sigma^2} \\
&=: a\mathbb{E}e^{\eta\|u_{t-1}\|_2^2} + b \qquad (4.82)
\end{aligned}$$

with $a := (1-s^2/2)e^{\frac{1}{2}mJ\eta\delta^2\sigma^2}$ and $b := se^{\frac{1}{2}mJ\eta\delta^2\sigma^2}$. Plugging (4.82) into (4.76), we have $\mathbb{E}e^{\eta\|u_t\|_2^2} \le a\mathbb{E}e^{\eta\|u_{t-1}\|_2^2} + b$. Next, similar to the argument in (4.21), iterating expectations yields $\mathbb{E}e^{\eta\|u_t\|_2^2} \le a^t\mathbb{E}(e^{\eta\|u_0\|_2^2}) + b(1+a+\ldots+a^t) = a^t + \frac{b(1-a^t)}{1-a} \le 1 + \frac{b}{1-a}$ where the last inequality holds if $a := (1-s^2/2)e^{mJ\eta\delta^2\sigma^2/2} < 1$. So we can now choose $\eta = \frac{-\log(1-s^2/2)}{mJ\delta^2\sigma^2}$, which satisfies $\eta\delta^2\sigma^2 \in [0,1/2]$ as required from before. Indeed, due to $m,J \ge 1$ and $s^2 = \frac{5}{9Jm} \le \frac{5}{9}$, we have $\eta\delta^2\sigma^2 = \frac{-\log(1-s^2/2)}{mJ} \le -\log(1-\frac{5}{18}) < \frac{1}{2}$. Then we get $a = (1-\frac{1}{2}s^2)^{1/2}$ and $b = s(1-\frac{1}{2}s^2)^{-1/2}$. It follows from (4.75) and $s^2 = \frac{5}{9mJ}$ that

$$\begin{aligned}
\mathrm{P}(\|u_t\|_2^2 \ge \alpha) &\le e^{-\eta\alpha}\Big(1+\frac{b}{1-a}\Big) = \exp\Big(\frac{\alpha\log(1-s^2/2)}{mJ\delta^2\sigma^2}\Big)\Big(1+\frac{s(1-\frac{1}{2}s^2)^{-1/2}}{1-\sqrt{1-s^2/2}}\Big) \\
&\le \exp\Big(\frac{-\alpha s^2}{2mJ\delta^2\sigma^2}\Big)\Big(1+\frac{s(1-\frac{1}{2}s^2)^{-1/2}+s}{s^2/2}\Big) \qquad (\text{since } \log(1+x)\le x) \\
&= \exp\Big(\frac{-\alpha s^2}{2mJ\delta^2\sigma^2}\Big)\Big(1+2\frac{(1-\frac{1}{2}s^2)^{-1/2}+1}{s}\Big) \\
&= \exp\Big(-\frac{5\alpha}{18m^2J^2\delta^2\sigma^2}\Big)\Big[1+6\sqrt{\frac{mJ}{5}}\Big(1-\frac{5}{18mJ}\Big)^{-1/2}+6\sqrt{\frac{mJ}{5}}\Big] \\
&\le 7\sqrt{mJ}\exp\Big(-\frac{\alpha}{4m^2J^2\delta^2\sigma^2}\Big)
\end{aligned}$$

where $c > 0$ is an absolute constant. Pick $\alpha = 4m^2J^2\delta^2\sigma^2\log(N_0^p)$ to get

$$\mathrm{P}\Big(\|u_t\|_2^2 \ge 4pm^2J^2\delta^2\sigma^2\log N_0\Big) \le 7\sqrt{mJ}N_0^{-p}. \qquad (4.83)$$

From (4.83) we can first conclude, by setting $t = N_0$ and using the fact $u_{N_0} = Xw - Xq$, that

$$\mathrm{P}\Big(\|Xw - Xq\|_2^2 \geq 4pm^2J^2\delta^2\sigma^2 \log N_0 \Big) \leq \frac{7\sqrt{mJ}}{N_0^p}.$$

If the activation function φ is ξ-Lipschitz, then $\|\varphi(Xw) - \varphi(Xq)\|_2 \leq \xi \|Xw - Xq\|_2$ and thus

$$\mathrm{P}\Big(\|\varphi(Xw) - \varphi(Xq)\|_2^2 \geq 4pm^2J^2\xi^2\delta^2\sigma^2 \log N_0 \Big) \leq \frac{7\sqrt{mJ}}{N_0^p}.$$

Moreover, applying a union bound over t, yields

$$\mathrm{P}\Big(\max_{1 \leq t \leq N_0} \|u_t\|_2^2 \geq 4pm^2J^2\delta^2\sigma^2 \log N_0 \Big) \leq \frac{7\sqrt{mJ}}{N_0^{p-1}}.$$

□

4.9.2 Proof of Corollary 4.3.5

We first need to bound the tail probability of $\frac{|\langle X_t, u \rangle|}{\|X_t\|_2^2}$ as follows.

Lemma 4.9.3. *Let clustered data $X = [X_1, X_2, \ldots, X_{N_0}] \in \mathbb{R}^{m \times N_0}$ be defined as in* (4.30) *and suppose that*

$$n \sum_{i=1}^{d} (z_t^{(i)})^2 \leq \frac{m\sigma^2}{36}, \quad 1 \leq t \leq N_0. \tag{4.84}$$

Let $u \in \mathbb{R}^m$ be a unit vector. For $1 \leq t \leq N_0$ and $x \geq \frac{16}{9m\sigma^2} \sqrt{n \sum_{i=1}^{d} (z_t^{(i)})^2}$, we have

$$\mathrm{P}\Big(\frac{|\langle X_t, u \rangle|}{\|X_t\|_2^2} \geq x \Big) \leq 2\exp\Big(-\frac{1}{2\sigma^2} \Big[\frac{9m\sigma^2 x}{16} - \big(n \sum_{i=1}^{d} (z_t^{(i)})^2 \big)^{\frac{1}{2}} \Big]^2 \Big) + \exp(-c_1 m)$$

where $c_1 > 0$ is a constant.

Proof. X_t can be expressed as $X_t = Z_t + \sigma G$ where $Z_t := [z_t^{(1)} \mathbb{1}_n, z_t^{(2)} \mathbb{1}_n, \ldots, z_t^{(d)} \mathbb{1}_n]^\top$ and $G \sim$

$\mathcal{N}(0,I)$. Since $\|Z_t\|_2^2 = n\sum_{i=1}^{d}(z_t^{(i)})^2$, then by the triangle inequality and (4.84)

$$\|X_t\|_2 \geq \sigma\|G\|_2 - \|Z_t\|_2 \geq \sigma\|G\|_2 - \frac{\sigma\sqrt{m}}{6}. \tag{4.85}$$

By Theorem 3.1.1 in [29], for any $x \geq 0$, we have

$$\mathrm{P}\Big(\Big|\frac{1}{m}\|G\|_2^2 - 1\Big| \geq \max\{x, x^2\}\Big) \leq 2\exp(-c_0 m x^2)$$

where $c_0 > 0$ is a constant. Choosing $x = \frac{23}{144}$, one can get

$$\mathrm{P}\Big(\|G\|_2^2 \geq \frac{121m}{144}\Big) \geq 1 - \exp(-c_1 m) \quad \text{with } c_1 := \Big(\frac{23}{144}\Big)^2 c_0. \tag{4.86}$$

If follows from (4.86) and (4.85) that

$$\mathrm{P}\Big(\|X_t\|_2 \geq \frac{3\sigma\sqrt{m}}{4}\Big) \geq \mathrm{P}\Big(\|G\|_2 \geq \frac{11\sqrt{m}}{12}\Big) \geq 1 - \exp(-c_1 m). \tag{4.87}$$

For a fixed Z_t and u both in $\mathbb{R}^m$ with $\|u\|_2 = 1$, define $f(z) := \langle \sigma z + Z_t, u\rangle$. Then $|f(z) - f(y)| = \sigma|\langle z - y, u\rangle| \leq \sigma\|z - y\|_2$. It follows from Theorem 8.40 of [9] that

$$\mathrm{P}(|\langle X_t, u\rangle - \langle Z_t, u\rangle| \geq \alpha) = \mathrm{P}(|f(G) - \mathbb{E}f(G)| \geq \alpha) \leq 2\exp\Big(-\frac{\alpha^2}{2\sigma^2}\Big) \tag{4.88}$$

holds for all $\alpha \geq 0$. For $x \geq 0$, the tail probability can be bounded as follows:

$$\begin{aligned}
\mathrm{P}\Big(\frac{|\langle X_t,u\rangle|}{\|X_t\|_2^2} \geq x\Big) &= \mathrm{P}\Big(\frac{|\langle X_t,u\rangle|}{\|X_t\|_2^2} \geq x, \|X_t\|_2^2 \geq \frac{9m\sigma^2}{16}\Big) + \mathrm{P}\Big(\frac{|\langle X_t,u\rangle|}{\|X_t\|_2^2} \geq x, \|X_t\|_2^2 < \frac{9m\sigma^2}{16}\Big) \\
&\leq \mathrm{P}\Big(|\langle X_t,u\rangle| \geq \frac{9m\sigma^2 x}{16}, \|X_t\|_2^2 \geq \frac{9m\sigma^2}{16}\Big) + \mathrm{P}\Big(\|X_t\|_2^2 < \frac{9m\sigma^2}{16}\Big) \\
&\leq \mathrm{P}\Big(|\langle X_t,u\rangle| \geq \frac{9m\sigma^2 x}{16}\Big) + \mathrm{P}\Big(\|X_t\|_2^2 < \frac{9m\sigma^2}{16}\Big) \\
&\leq \mathrm{P}\Big(|\langle X_t,u\rangle - \langle Z_t,u\rangle| \geq \frac{9m\sigma^2 x}{16} - |\langle Z_t,u\rangle|\Big) + \mathrm{P}\Big(\|X_t\|_2^2 < \frac{9m\sigma^2}{16}\Big) \\
&\leq \mathrm{P}\Big(|\langle X_t,u\rangle - \langle Z_t,u\rangle| \geq \frac{9m\sigma^2 x}{16} - \|Z_t\|_2\Big) + \mathrm{P}\Big(\|X_t\|_2^2 < \frac{9m\sigma^2}{16}\Big).
\end{aligned}$$

If $\|Z_t\|_2 = \sqrt{n\sum_{i=1}^d (z_t^{(i)})^2} \leq \frac{9m\sigma^2 x}{16}$, then, by (4.87) and (4.88), we obtain

$$\begin{aligned}
\mathrm{P}\Big(\frac{|\langle X_t,u\rangle|}{\|X_t\|_2^2} \geq x\Big) &\leq 2\exp\Big(-\frac{1}{2\sigma^2}\Big(\frac{9m\sigma^2 x}{16} - \|Z_t\|_2\Big)^2\Big) + \mathrm{P}\Big(\|X_t\|_2^2 < \frac{9m\sigma^2}{16}\Big) \\
&\leq 2\exp\Big(-\frac{1}{2\sigma^2}\Big[\frac{9m\sigma^2 x}{16} - \Big(n\sum_{i=1}^d (z_t^{(i)})^2\Big)^{\frac{1}{2}}\Big]^2\Big) + \exp(-c_1 m).
\end{aligned}$$

□

Now we are ready to prove Corollary 4.3.5.

Proof. Since $J = 1 + (d\sigma^2)^{-1}\max_{1\leq t\leq N_0}\sum_{i=1}^d (z_t^{(i)})^2 \leq 1 + \frac{\log N_0}{36m}$ and $m \geq \log N_0$, we have $\sum_{i=1}^d (z_t^{(i)})^2 \leq \frac{\sigma^2}{36n}\log N_0 \leq \frac{m\sigma^2}{36n}$ for all t. Moreover, Lemma 4.9.3 and $m \geq \frac{2}{c_1}\log N_0$ indicate that, for $u \in \mathbb{S}^{m-1}$, we have

$$\mathrm{P}\Big(\frac{|\langle X_t,u\rangle|}{\|X_t\|_2^2} \geq x\Big) \leq 2\exp\Big(-\frac{1}{2}\Big(\frac{9m\sigma x}{16} - \frac{\sqrt{\log N_0}}{6}\Big)^2\Big) + \frac{1}{N_0^2} \tag{4.89}$$

where $x \geq \frac{8}{27m\sigma}\sqrt{\log N_0}$. Now, we consider the t-th iteration of (4.12) and let $\mathscr{S}_{t-1}$ be the event $\|u_{t-1}\|_2^2 \leq 4pm^2J^2\delta^2\sigma^2\log N_0$. By Theorem 4.3.4, we have

$$\mathrm{P}(\mathscr{S}_{t-1}) \geq 1 - \frac{7\sqrt{mJ}}{N_0^p}. \tag{4.90}$$

Conditioning on $\mathscr{S}_{t-1}$ and applying (4.89) with $x = \frac{4\sqrt{\log N_0}}{m\sigma}$, we have

$$\begin{aligned}
&\mathrm{P}\Big(\Big|w_t + \frac{\langle X_t, u_{t-1}\rangle}{\|X_t\|_2^2}\Big| > \|w\|_\infty + 8\delta J\sqrt{p}\log N_0 \mid \mathscr{S}_{t-1}\Big) \\
&\le \mathrm{P}\Big(\Big|\frac{\langle X_t, u_{t-1}\rangle}{\|X_t\|_2^2\|u_{t-1}\|_2}\Big| \ge \frac{8\delta J\sqrt{p}\log N_0}{\|u_{t-1}\|_2} \mid \mathscr{S}_{t-1}\Big) \\
&\le \mathrm{P}\Big(\Big|\frac{\langle X_t, u_{t-1}\rangle}{\|X_t\|_2^2\|u_{t-1}\|_2}\Big| \ge \frac{4\sqrt{\log N_0}}{m\sigma} \mid \mathscr{S}_{t-1}\Big) \\
&\le \frac{3}{N_0^2}. \qquad (4.91)
\end{aligned}$$

Combining (4.90) and (4.91), we obtain

$$\mathrm{P}\Big(\Big|w_t + \frac{\langle X_t, u_{t-1}\rangle}{\|X_t\|_2^2}\Big| \le \|w\|_\infty + 8\delta J\sqrt{p}\log N_0, \mathscr{S}_{t-1}\Big) \ge 1 - \frac{7\sqrt{mJ}}{N_0^p} - \frac{3}{N_0^2}.$$

By a union bound over t, we obtain (4.31). □

4.10 Theoretical Analysis for Sparse GPFQ

In this section, we will show that Theorem 4.4.1 and Theorem 4.4.2 (restated here for convenience) hold.

Theorem 4.4.1: Under the conditions of Theorem 4.3.1, we have the following.

(a) Quantizing w using (4.37) with the alphabet $\mathscr{A}$ in (4.3), we have

$$\mathrm{P}\Big(\|Xw - Xq\|_2^2 \le \frac{r^2(2\lambda+\delta)^2}{s^2}\log N_0\Big) \ge 1 - \frac{1}{N_0^2}\Big(2 + \frac{1}{\sqrt{1-s^2}}\Big).$$

(b) Quantizing w using (4.38) with the alphabet $\widetilde{\mathscr{A}}$ in (4.5), we have

$$\mathrm{P}\Big(\|Xw - Xq\|_2^2 \le \frac{r^2\max\{2\lambda,\delta\}^2}{s^2}\log N_0\Big) \ge 1 - \frac{1}{N_0^2}\Big(2 + \frac{1}{\sqrt{1-s^2}}\Big).$$

Theorem 4.4.2: Under the assumptions of Theorem 4.3.4, the followings inequalities hold.

(a) Quantizing w using (4.37) with the alphabet $\mathscr{A}$ in (4.3), we have

$$\mathrm{P}\Big(\|Xw - Xq\|_2^2 \geq 4pm^2J^2(2\lambda+\delta)^2\sigma^2 \log N_0\Big) \leq \frac{7\sqrt{mJ}}{N_0^p}.$$

(b) Quantizing w using (4.38) with the alphabet $\widetilde{\mathscr{A}}$ in (4.5), we have

$$\mathrm{P}\Big(\|Xw - Xq\|_2^2 \geq 4pm^2J^2\max\{2\lambda,\delta\}^2\sigma^2 \log N_0\Big) \leq \frac{7\sqrt{mJ}}{N_0^p}.$$

Note that the difference between the sparse GPFQ and the GPFQ in (4.12) is the usage of thresholding functions. So the key point is to adapt Lemma 4.6.3 for those changes.

4.10.1 Sparse GPFQ with Soft Thresholding

We first focus on the error analysis for (4.37) which needs the following lemmata.

Lemma 4.10.1. *Let $\mathscr{A}$ be one of the alphabets defined in* (4.4) *with step size $\delta > 0$, and the largest element q_{max}. Let $\theta_t := \angle(X_t, u_{t-1})$ be the angle between X_t and u_{t-1}. Suppose that $w \in \mathbb{R}^{N_0}$ satisfies $\|w\|_\infty \leq q_{\max}$, and consider the quantization scheme given by* (4.37)*. Then, for $t = 1,2,\ldots,N_0$, we have*

$$\|u_t\|_2^2 - \|u_{t-1}\|_2^2 \leq \begin{cases} \frac{(2\lambda+\delta)^2}{4}\|X_t\|_2^2 - \|u_{t-1}\|_2^2\cos^2\theta_t & \text{if } \left|w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\right| \leq q_{max}+\lambda, \\ 0 & \text{otherwise}. \end{cases} \tag{4.92}$$

Proof. By applying exactly the same argument as in Lemma 4.6.3, one can get

$$\|P_{X_t}(u_t)\|_2^2 = \left(w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t - q_t\right)^2\|X_t\|_2^2. \tag{4.93}$$

and

$$\Big(w_t+\frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t-q_t\Big)^2-\Big(\frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\Big)^2=\underbrace{\Big(w_t+\frac{2\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t-q_t\Big)}_{\text{(I)}}\underbrace{(w_t-q_t)}_{\text{(II)}},$$

where $|w_t|\le q_{\max}$ and $q_t=\mathscr{Q}\circ s_\lambda\Big(w_t+\frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\Big)$. We proceed by going through the cases.

First, if $\Big(w_t+\frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\Big)>q_{\max}+\lambda$, then $q_t=q_{\max}$ and thus $\lambda\le q_t-w_t+\lambda<\frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t$. So (I) $>w_t+2(q_t-w_t+\lambda)-q_t=q_t-w_t+2\lambda\ge 2\lambda$ and (II) ≤ 0. Moreover, if $\Big(w_t+\frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\Big)<-q_{\max}-\lambda$, then $q_t=-q_{\max}$ and $\frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t<q_t-w_t-\lambda\le-\lambda$. Hence, (I) $<w_t+2(q_t-w_t-\lambda)-q_t=q_t-w_t-2\lambda\le-2\lambda$ and (II) ≥ 0. It follows that

$$\Big(w_t+\frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t-q_t\Big)^2\le\Big(\frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\Big)^2 \tag{4.94}$$

when $\Big|w_t+\frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\Big|>q_{\max}+\lambda$.

Now, assume that $\Big|w_t+\frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\Big|\le q_{\max}+\lambda$. In this case, let $v_t:=s_\lambda\Big(w_t+\frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\Big)$. Then $|v_t|\le q_{\max}$ and $\Big|w_t+\frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t-v_t\Big|\le\lambda$. Since $q_t=\mathscr{Q}(v_t)$, we obtain

$$\begin{aligned}\Big(w_t+\frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t-q_t\Big)^2&=\Big|w_t+\frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t-v_t+v_t-q_t\Big|^2\\&\le\Big(\Big|w_t+\frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t-v_t\Big|+|v_t-q_t|\Big)^2\\&\le\Big(\lambda+\frac{\delta}{2}\Big)^2.\end{aligned} \tag{4.95}$$

Applying (4.94) and (4.95) to (4.93), one can get

$$\|P_{X_t}(u_t)\|_2^2\le\begin{cases}\frac{(2\lambda+\delta)^2}{4}\|X_t\|_2^2 & \text{if } \Big|w_t+\frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\Big|\le q_{\max}+\lambda,\\ \|u_{t-1}\|_2^2\cos^2\theta_t & \text{otherwise.}\end{cases} \tag{4.96}$$

Again, by the same discussion after (4.53) in Lemma 4.6.3, we have

$$\|u_t\|_2^2 - \|u_{t-1}\|_2^2 = \|P_{X_t}(u_t)\|_2^2 - \|u_{t-1}\|_2^2 \cos^2\theta_t.$$

Replacing $\|P_{X_t}(u_t)\|_2^2$ with its upper bounds in (4.96), we obtain (4.92). □

Now we are ready to prove Theorem 4.4.1 as follows.

Proof. The only difference between Lemma 4.6.3 and its analogue Lemma 4.10.1 is that δ^2 in Lemma 4.6.3 is replaced by $(2\lambda+\delta)^2$. Note that Lemma 4.6.3 was used in the proof of both Theorem 4.3.1 and Theorem 4.3.4 in which δ^2 serves as a coefficient. Hence, by substituting δ^2 with $(2\lambda+\delta)^2$, every step in the proof still works and thus Theorem 4.4.1 holds. □

4.10.2 Sparse GPFQ with Hard Thresholding

Now we navigate to the error analysis for (4.38). Again, Lemma 4.6.3 is altered as follows.

Lemma 4.10.2. *Let $\widetilde{\mathscr{A}}$ be one of the alphabets defined in* (4.6) *with step size $\delta > 0$, the largest element q_{max}, and threshold $\lambda \in (0, q_{\max})$. Let $\theta_t := \angle(X_t, u_{t-1})$ be the angle between X_t and u_{t-1}. Suppose that $w \in \mathbb{R}^{N_0}$ satisfies $\|w\|_\infty \le q_{\max}$, and consider the quantization scheme given by* (4.38)*. Then, for $t = 1, 2, \ldots, N_0$, we have*

$$\|u_t\|_2^2 - \|u_{t-1}\|_2^2 \le \begin{cases} \frac{\max\{2\lambda,\delta\}^2}{4}\|X_t\|_2^2 - \|u_{t-1}\|_2^2\cos^2\theta_t & \text{if } \left|w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t\right| \le q_{max}, \\ 0 & \text{otherwise}. \end{cases} \tag{4.97}$$

Proof. By applying exactly the same argument as in Lemma 4.6.3, we obtain

$$\|P_{X_t}(u_t)\|_2^2 = \left(w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2}\cos\theta_t - q_t\right)^2 \|X_t\|_2^2. \tag{4.98}$$

where $|w_t| \leq q_{\max}$ and $q_t = \mathscr{Q} \circ h_\lambda \Big(w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2} \cos\theta_t \Big)$. Due to $\lambda \in (0, q_{\max})$, we have $\mathscr{Q} \circ h_\lambda(z) = \mathscr{Q}(z)$ for $|z| > q_{\max}$. Thus, it follows from the discussion in Lemma 4.6.3 that

$$\Big(w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2} \cos\theta_t - q_t \Big)^2 \leq \Big(\frac{\|u_{t-1}\|_2}{\|X_t\|_2} \cos\theta_t \Big)^2 \tag{4.99}$$

when $\Big| w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2} \cos\theta_t \Big| > q_{\max}$.

Now, assume that $\Big| w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2} \cos\theta_t \Big| \leq q_{\max}$. In this case, because the argument of $\mathscr{Q}$ lies in the active range of $\mathscr{A}$, we obtain

$$\Big(w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2} \cos\theta_t - q_t \Big)^2 \leq \max\Big\{ \lambda, \frac{\delta}{2} \Big\}^2. \tag{4.100}$$

Applying (4.99) and (4.100) to (4.98), one can get

$$\|P_{X_t}(u_t)\|_2^2 \leq \begin{cases} \frac{\max\{2\lambda, \delta\}^2}{4} \|X_t\|_2^2 & \text{if } \Big| w_t + \frac{\|u_{t-1}\|_2}{\|X_t\|_2} \cos\theta_t \Big| \leq q_{\max}, \\ \|u_{t-1}\|_2^2 \cos^2\theta_t & \text{otherwise.} \end{cases} \tag{4.101}$$

Again, by the same discussion after (4.53) in Lemma 4.6.3, we have

$$\|u_t\|_2^2 - \|u_{t-1}\|_2^2 = \|P_{X_t}(u_t)\|_2^2 - \|u_{t-1}\|_2^2 \cos^2\theta_t.$$

Replacing $\|P_{X_t}(u_t)\|_2^2$ with its upper bounds in (4.101), we obtain (4.97). □

The proof of Theorem 4.4.2 is given as follows.

Proof. The only difference between Lemma 4.6.3 and its analogue Lemma 4.10.2 is that δ^2 in Lemma 4.6.3 is replaced by $\max\{2\lambda + \delta\}^2$. Note that Lemma 4.6.3 was used in the proof of both Theorem 4.3.1 and Theorem 4.3.4 in which δ^2 serves as a coefficient. Hence, by substituting δ^2 with $\max\{2\lambda + \delta\}^2$, it is not hard to verify that Theorem 4.4.2 holds. □

4.11 Acknowledgements

We thank Eric Lybrand for stimulating discussions on the topics of this paper. This work was supported in part by National Science Foundation Grant DMS-2012546. This chapter, in full, is joint work with Yixuan Zhou, Rayan Saab, and has been published in the SIAM Journal on Mathematics of Data Science (SIMODS), 2023. The dissertation author was the primary investigator and author of this material.

References

[1] Ron Banner, Yury Nahshan, and Daniel Soudry. "Post training 4-bit quantization of convolutional networks for rapid-deployment". In: *Advances in Neural Information Processing Systems* 32 (2019).

[2] Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. "Zeroq: A novel zero shot quantization framework". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 13169–13178.

[3] Yoni Choukroun, Eli Kravchik, Fan Yang, and Pavel Kisilev. "Low-bit Quantization of Neural Networks for Efficient Inference." In: *ICCV Workshops*. 2019, pp. 3009–3018.

[4] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. "Binaryconnect: Training deep neural networks with binary weights during propagations". In: *Advances in neural information processing systems*. 2015, pp. 3123–3131.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[6] Lei Deng, Guoqi Li, Song Han, Luping Shi, and Yuan Xie. “Model compression and hardware acceleration for neural networks: A comprehensive survey”. In: *Proceedings of the IEEE* 108.4 (2020), pp. 485–532.

[7] Zhen Dong, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. “Hawq: Hessian aware quantization of neural networks with mixed-precision”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 293–302.

[8] Jun Fang, Ali Shafiee, Hamzah Abdel-Aziz, David Thorsley, Georgios Georgiadis, and Joseph H Hassoun. “Post-training piecewise linear quantization for deep neural networks”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 69–86.

[9] Simon Foucart and Holger Rauhut. “An invitation to compressive sensing”. In: *A mathematical introduction to compressive sensing*. Springer, 2013, pp. 1–39.

[10] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. “A survey of quantization methods for efficient neural network inference”. In: *arXiv preprint arXiv:2103.13630* (2021).

[11] Yunhui Guo. “A survey on methods and theories of quantized neural networks”. In: *arXiv preprint arXiv:1808.04752* (2018).

[12] Song Han, Huizi Mao, and William J Dally. “Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding”. In: *arXiv preprint arXiv:1510.00149* (2015).

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[14] Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. “Improving post training neural quantization: Layer-wise calibration and integer programming”. In: *arXiv preprint arXiv:2006.10518* (2020).

[15] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. “Quantization and training of neural networks for efficient integer-arithmetic-only inference”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2704–2713.

[16] Raghuraman Krishnamoorthi. “Quantizing deep convolutional networks for efficient inference: A whitepaper”. In: *arXiv preprint arXiv:1806.08342* (2018).

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012), pp. 1097–1105.

[18] Jun Haeng Lee, Sangwon Ha, Saerom Choi, Won-Jo Lee, and Seungwon Lee. “Quantization for rapid deployment of deep neural networks”. In: *arXiv preprint arXiv:1810.05488* (2018).

[19] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. “Brecq: Pushing the limit of post-training quantization by block reconstruction”. In: *International Conference on Learning Representations* (2021).

[20] Yuang Liu, Wei Zhang, and Jun Wang. “Zero-shot Adversarial Quantization”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 1512–1521.

[21] Christos Louizos, Matthias Reisser, Tijmen Blankevoort, Efstratios Gavves, and Max Welling. “Relaxed Quantization for Discretized Neural Networks”. In: *International Conference on Learning Representations*. 2019.

[22] Eric Lybrand and Rayan Saab. “A Greedy Algorithm for Quantizing Neural Networks”. In: *Journal of Machine Learning Research* 22.156 (2021), pp. 1–38.

[23] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen B. "Up or down? adaptive rounding for post-training quantization". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 7197–7206.

[24] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. "Data-free quantization through weight equalization and bias correction". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1325–1334.

[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32 (2019), pp. 8026–8037.

[26] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *International Conference on Learning Representations* (2015).

[27] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[28] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 6105–6114.

[29] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*. Vol. 47. Cambridge university press, 2018.

[30] Peisong Wang, Qiang Chen, Xiangyu He, and Jian Cheng. "Towards accurate post-training network quantization via bit-split and stitching". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9847–9856.

[31] Peisong Wang, Xiangyu He, Gang Li, Tianli Zhao, and Jian Cheng. “Sparsity-inducing binarized neural networks”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 2020, pp. 12192–12199.

[32] Shoukai Xu, Haokun Li, Bohan Zhuang, Jing Liu, Jiezhang Cao, Chuangrun Liang, and Mingkui Tan. “Generative low-bitwidth data free quantization”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 1–17.

[33] Dongqing Zhang, Jiaolong Yang, Dongqiangzi Ye, and Gang Hua. “Lq-nets: Learned quantization for highly accurate and compact deep neural networks”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 365–382.

[34] Ritchie Zhao, Yuwei Hu, Jordan Dotzel, Chris De Sa, and Zhiru Zhang. “Improving neural network quantization without retraining using outlier channel splitting”. In: *International conference on machine learning*. PMLR. 2019, pp. 7543–7552.

[35] Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen. “Incremental network quantization: Towards lossless cnns with low-precision weights”. In: *arXiv preprint arXiv:1702.03044* (2017).

[36] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. “Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients”. In: *arXiv preprint arXiv:1606.06160* (2016).

Chapter 5

A Stochastic Algorithm and its Error Analysis for Neural Network Quantization

Quantization is a widely used compression method that effectively reduces redundancies in over-parameterized neural networks. However, existing quantization techniques for deep neural networks often lack a comprehensive error analysis due to the presence of non-convex loss functions and nonlinear activations. In this paper, we propose a stochastic algorithm for quantizing the weights of fully trained neural networks. Our approach leverages a greedy path-following mechanism in combination with a stochastic quantizer. Importantly, we establish, for the first time, full-network error bounds, under an infinite alphabet condition and minimal assumptions on the weights and input data. As an application of this result, we prove that when quantizing a multi-layer network having Gaussian weights, the relative quantization error exhibits a linear decay as the degree of over-parametrization increases. Furthermore, we demonstrate that it is possible to achieve error bounds equivalent to those obtained in the infinite alphabet case, using on the order of a mere $\log\log N$ bits, where N represents the largest width, i.e., largest number of neurons, of a layer.

5.1 Introduction

Deep neural networks (DNNs) have shown impressive performance in a variety of areas including computer vision and natural language processing among many others. However, highly overparameterized DNNs require a significant amount of memory to store their associated weights, activations, and – during training – gradients. As a result, in recent years, there has been an interest in model compression techniques, including quantization, pruning, knowledge distillation, and low-rank decomposition [27, 12, 7, 15, 16]. Neural network quantization, in particular, utilizes significantly fewer bits to represent the weights of DNNs. This substitution of original, say, 32-bit floating-point operations with more efficient low-bit operations has the potential to significantly reduce memory usage and accelerate inference time while maintaining minimal loss in accuracy. Quantization methods can be categorized into two classes [22]: quantization-aware training and post-training quantization. Quantization-aware training substitutes floating-point weights with low-bit representations during the training process, while post-training quantization quantizes network weights only after the training is complete.

To achieve high-quality empirical results, quantization-aware training methods, such as those in [8, 6, 35, 10, 21, 37, 40], often require significant time for retraining and hyper-parameter tuning using the entire training dataset. This can make them impractical for resource-constrained scenarios. Furthermore, it can be challenging to rigorously analyze the associated error bounds as quantization-aware training is an integer programming problem with a non-convex loss function, making it NP-hard in general. In contrast, post-training quantization algorithms, such as [9, 36, 23, 38, 20, 26, 39, 25, 14], require only a small amount of training data, and recent research has made strides in obtaining quantization error bounds for some of these algorithms [23, 38, 25] in the context of shallow networks.

In this paper, we focus on this type of network quantization and its theoretical analysis, proposing a stochastic quantization technique and obtaining theoretical guarantees on its performance, even in the context of deep networks.

5.1.1 Related work

In this section, we provide a summary of relevant prior results concerning a specific post-training quantization algorithm, which forms the basis of our present work. To make our discussion more precise, let $X \in \mathbb{R}^{m\times N_0}$ and $w \in \mathbb{R}^{N_0}$ represent the input data and a neuron in a single-layer network, respectively. Our objective is to find a mapping, also known as a *quantizer*, $\mathscr{Q} : \mathbb{R}^{N_0} \to \mathscr{A}^{N_0}$ such that $q = Q(w) \in \mathscr{A}^{N_0}$ minimizes $\|Xq - Xw\|_2$. Even in this simplified context, since $\mathscr{A}$ is a finite discrete set, this optimization problem is an integer program and therefore NP-hard in general. Nevertheless, if one can obtain good approximate solutions to this optimization problem, with theoretical error guarantees, then those guarantees can be combined with the fact that most neural network activation functions are Lipschitz, to obtain error bounds on entire (single) layers of a neural network.

Recently, Lybrand and Saab [23] proposed and analyzed a greedy algorithm, called *greedy path following quantization* (GPFQ), to approximately solve the optimization problem outlined above. Their analysis was limited to the ternary alphabet $\mathscr{A} = \{0, \pm 1\}$ and a single-layer network with Gaussian random input data. Zhang, Zhou, and Saab [38] then extended GPFQ to more general input distributions and larger alphabets, and they introduced variations that promoted pruning of weights. Among other results, they proved that if the input data X is either bounded or drawn from a mixture of Gaussians, then the relative square error of quantizing a generic neuron w satisfies

$$\frac{\|Xw - Xq\|_2^2}{\|Xw\|_2^2} \lesssim \frac{m \log N_0}{N_0} \tag{5.1}$$

with high probability. Extensive numerical experiments in [38] also demonstrated that GPFQ, with 4 or 5 bit alphabets, can achieve less than 1% loss in Top-1 and Top-5 accuracy on common neural network architectures. Subsequently, [25] introduced a different algorithm that involves a deterministic preprocessing step on w that allows quantizing DNNs via *memoryless scalar quantization* (MSQ) while preserving the same error bound in (5.1). This algorithm is more computationally intensive than those of [23, 38] but does not require hyper-parameter tuning for

selecting the alphabet step-size.

5.1.2 Contributions and organization

In spite of recent progress in developing computationally efficient algorithms with rigorous theoretical guarantees, all technical proofs in [23, 38, 25] only apply for a single-layer of a neural network with certain assumed input distributions. This limitation naturally comes from the fact that a random input distribution and a deterministic quantizer lead to activations (i.e., outputs of intermediate layers) with dependencies, whose distribution is usually intractable after passing through multiple layers and nonlinearities.

To overcome this main obstacle to obtaining theoretical guarantees for multiple layer neural networks, in Section 5.2, we propose a new stochastic quantization framework, called stochastic path following quantization (SPFQ), which introduces randomness into the quantizer. We show that SPFQ admits an interpretation as a two-phase algorithm consisting of a data-alignment phase and a quantization phase. This allows us to propose two variants given by Algorithm 5 and Algorithm 6, which involve different data alignment strategies that are amenable to analysis.

In Section 5.3, we prove the first error bounds for quantizing an entire L-layer neural network Φ, under an infinite alphabet condition and minimal assumptions on the weights and input data X. To illustrate the use of our results, we show that if the weights of Φ are standard Gaussian random variables, then, with high probability, the quantized neural network $\widetilde{\Phi}$ satisfies

$$\frac{\|\Phi(X)-\widetilde{\Phi}(X)\|_F^2}{\mathbb{E}_\Phi\|\Phi(X)\|_F^2} \lesssim \frac{m(\log N_{\max})^{L+1}}{N_{\min}} \tag{5.2}$$

where we take the expectation $\mathbb{E}_\Phi$ with respect to the weights of Φ, and $N_{\min}$, $N_{\max}$ represent the minimum and maximum layer width of Φ respectively. We can regard the relative error bound in (5.2) as a natural generalization of (5.1).

In Section 5.4, we consider the finite alphabet case under the random network hypothesis.

Denoting by N_i the number of neurons in the i-th layer, we show that it suffices to use $b \leq C \log\log \max\{N_{i-1}, N_i\}$ bits to quantize the i-th layer while guaranteeing the same error bounds as in the infinite alphabet case.

It is worth noting that we assume that Φ is equipped with ReLU activation functions, i.e. $\max\{0,x\}$, throughout this paper. This assumption is only made for convenience and concreteness, and we remark that the non-linearities can be replaced by any 1-Lipschitz functions without changing our results, except for the values of constants.

Finally, we empirically test the developed method in Section 5.5, by quantizing the weights of several neural network architectures with ImageNet dataset, presenting only minor loss of accuracy compared to unquantized models.

5.2 Stochastic Quantization Algorithm

In this section, we start with the notation that will be used throughout this paper and then introduce our stochastic quantization algorithm, and show that it can be viewed as a two-stage algorithm. This in turn will simplify its analysis.

5.2.1 Notation and Preliminaries

We denote various positive absolute constants by C, c. We use $a \lesssim b$ as shorthand for $a \leq Cb$, and $a \gtrsim b$ for $a \geq Cb$. For any matrix $A \in \mathbb{R}^{m\times n}$, $\|A\|_{\max}$ denotes $\max_{i,j} |A_{ij}|$.

Quantization

An L-layer perceptron, $\Phi : \mathbb{R}^{N_0} \to \mathbb{R}^{N_L}$, acts on a vector $x \in \mathbb{R}^{N_0}$ via

$$\Phi(x) := \varphi^{(L)} \circ A^{(L)} \circ \cdots \circ \varphi^{(1)} \circ A^{(1)}(x) \tag{5.3}$$

where each $\varphi^{(i)} : \mathbb{R}^{N_i} \to \mathbb{R}^{N_i}$ is an activation function acting entrywise, and $A^{(i)} : \mathbb{R}^{N_{i-1}} \to \mathbb{R}^{N_i}$ is an affine map given by $A^{(i)}(z) := W^{(i)\top} z + b^{(i)}$. Here, $W^{(i)} \in \mathbb{R}^{N_{i-1}\times N_i}$ is a weight matrix and

$b^{(i)} \in \mathbb{R}^{N_i}$ is a bias vector. Since $w^\top x + b = \langle (w,b),(x,1)\rangle$, the bias term $b^{(i)}$ can simply be treated as an extra row to the weight matrix $W^{(i)}$, so we will henceforth ignore it. For theoretical analysis, we focus on infinite *mid-tread* alphabets, with step-size δ, i.e., alphabets of the form

$$\mathscr{A} = \mathscr{A}_\infty^\delta := \{\pm k\delta : k \in \mathbb{Z}\} \tag{5.4}$$

and their finite versions, mid-tread alphabets of the form

$$\mathscr{A} = \mathscr{A}_K^\delta := \{\pm k\delta : 0 \le k \le K, k \in \mathbb{Z}\}. \tag{5.5}$$

Given $\mathscr{A} = \mathscr{A}_\infty^\delta$, the associated *stochastic scalar quantizer* $\mathscr{Q}_{\text{StocQ}} : \mathbb{R} \to \mathscr{A}$ randomly rounds every $z \in \mathbb{R}$ to either the minimum or maximum of the interval $[k\delta,(k+1)\delta]$ containing it, in such a way that $\mathbb{E}(\mathscr{Q}_{\text{StocQ}}(z)) = z$. Specifically, we define

$$\mathscr{Q}_{\text{StocQ}}(z) := \begin{cases} \lfloor \frac{z}{\delta} \rfloor \delta & \text{with probability } p \\ \big(\lfloor \frac{z}{\delta} \rfloor + 1\big)\delta & \text{with probability } 1-p \end{cases} \tag{5.6}$$

where $p = 1 - \frac{z}{\delta} + \lfloor \frac{z}{\delta} \rfloor$. If instead of the infinite alphabet, we use $\mathscr{A} = \mathscr{A}_K^\delta$, then whenever $|z| \le K\delta$, $\mathscr{Q}_{\text{StocQ}}(z)$ is defined via (5.6) while $\mathscr{Q}_{\text{StocQ}}(z)$ is assigned $-K\delta$ and $K\delta$ if $z < -K\delta$ and $z > K\delta$ respectively.

Orthogonal projections

Given a subspace $S \subseteq \mathbb{R}^m$, we denote by $S^\perp$ its orthogonal complement in $\mathbb{R}^m$, and by P_S the orthogonal projection of $\mathbb{R}^m$ onto S. In particular, if $z \in \mathbb{R}^m$ is a nonzero vector, then we use P_z and $P_{z^\perp}$ to represent orthogonal projections onto $\text{span}(z)$ and $\text{span}(z)^\perp$ respectively. Hence, for any $x \in \mathbb{R}^m$, we have

$$P_z(x) = \frac{\langle z,x\rangle z}{\|z\|_2^2}, \quad x = P_z(x) + P_{z^\perp}(x), \quad \text{and} \quad \|x\|_2^2 = \|P_z(x)\|_2^2 + \|P_{z^\perp}(x)\|_2^2. \tag{5.7}$$

Throughout this paper, we will also use P_z and $P_{z^\perp}$ to denote the associated matrix representations satisfying

$$P_z x = \frac{zz^\top}{\|z\|_2^2} x \quad \text{and} \quad P_{z^\perp} x = \Big(I - \frac{zz^\top}{\|z\|_2^2}\Big) x. \tag{5.8}$$

Convex orders

We now introduce the concept of *convex order* (see, e.g., [32]), which will be heavily used in our analysis.

Definition 5.2.1. *Let X, Y be n-dimensional random vectors such that*

$$\mathbb{E} f(X) \leq \mathbb{E} f(Y) \tag{5.9}$$

holds for all convex functions $f : \mathbb{R}^n \to \mathbb{R}$, provided the expectations exist. Then X is said to be smaller than Y in the convex order*, denoted by $X \leq_{\text{cx}} Y$.*

For $i = 1, 2, \ldots, n$, define functions $\phi_i(x) := x_i$ and $\psi_i(x) := -x_i$. Since both $\phi_i(x)$ and $\psi_i(x)$ are convex, substituting them into (5.9) yields $\mathbb{E}X_i = \mathbb{E}Y_i$ for all i. Therefore, we obtain

$$X \leq_{\text{cx}} Y \implies \mathbb{E}X = \mathbb{E}Y. \tag{5.10}$$

Clearly, according to Definition 5.2.1, $X \leq_{\text{cx}} Y$ only depends on the respective distributions of X and Y. It can be easily seen that the relation $\leq_{\text{cx}}$ satisfies reflexivity and transitivity. In other words, one has $X \leq_{\text{cx}} X$ and that if $X \leq_{\text{cx}} Y$ and $Y \leq_{\text{cx}} Z$, then $X \leq_{\text{cx}} Z$. The convex order defined in Definition 5.2.1 is also called *mean-preserving spread* [31, 24], which is a special case of *second-order stochastic dominance* [17, 18, 32], see Section 5.6 for details.

5.2.2 SPFQ

We start with a data set $X \in \mathbb{R}^{m \times N_0}$ with (vectorized) data stored as rows and a pretrained neural network Φ with weight matrices $W^{(i)} \in \mathbb{R}^{N_{i-1} \times N_i}$ having neurons as their columns. Let

$\Phi^{(i)}$, $\widetilde{\Phi}^{(i)}$ denote the original and quantized neural networks up to layer i respectively so that, for example, $\Phi^{(i)}(x) := \varphi^{(i)} \circ W^{(i)} \circ \cdots \circ \varphi^{(1)} \circ W^{(1)}(x)$. Assuming the first $i-1$ layers have been quantized, define the *activations* from $(i-1)$-th layer as

$$X^{(i-1)} := \Phi^{(i-1)}(X) \in \mathbb{R}^{m \times N_{i-1}} \quad \text{and} \quad \widetilde{X}^{(i-1)} := \widetilde{\Phi}^{(i-1)}(X) \in \mathbb{R}^{m \times N_{i-1}}, \tag{5.11}$$

which also serve as input data for the i-th layer. For each neuron $w \in \mathbb{R}^{N_{i-1}}$ in layer i, our goal is to construct a quantized vector $q \in \mathscr{A}^{N_{i-1}}$ such that

$$\widetilde{X}^{(i-1)} q = \sum_{t=1}^{N_{i-1}} q_t \widetilde{X}_t^{(i-1)} \approx \sum_{t=1}^{N_{i-1}} w_t X_t^{(i-1)} = X^{(i-1)} w$$

where $X_t^{(i-1)}$, $\widetilde{X}_t^{(i-1)}$ are the t-th columns of $X^{(i-1)}$, $\widetilde{X}^{(i-1)}$. Following the GPFQ scheme in [23, 38], our algorithm selects q_t sequentially, for $t = 1, 2, \ldots, N_{i-1}$, so that the approximation error of the t-th iteration, denoted by

$$u_t := \sum_{j=1}^{t} w_j X_j^{(i-1)} - \sum_{j=1}^{t} q_j \widetilde{X}_j^{(i-1)} \in \mathbb{R}^m, \tag{5.12}$$

is well-controlled in the ℓ_2 norm. Specifically, assuming that the first $t-1$ components of q have been determined, the proposed algorithm maintains the error vector $u_{t-1} = \sum_{j=1}^{t-1} (w_j X_j^{(i-1)} - q_j \widetilde{X}_j^{(i-1)})$, and sets $q_t \in \mathscr{A}$ probabilistically depending on u_{t-1}, $X_t^{(i-1)}$, and $\widetilde{X}_t^{(i-1)}$. Note that (5.12) implies

$$u_t = u_{t-1} + w_t X_t^{(i-1)} - q_t \widetilde{X}_t^{(i-1)} \tag{5.13}$$

and using (5.7), one can get

$$\begin{aligned} c^* &:= \arg\min_{c\in\mathbb{R}} \|u_{t-1} + w_t X_t^{(i-1)} - c\widetilde{X}_t^{(i-1)}\|_2^2 \\ &= \arg\min_{c\in\mathbb{R}} \|P_{\widetilde{X}_t^{(i-1)}}(u_{t-1} + w_t X_t^{(i-1)}) - c\widetilde{X}_t^{(i-1)}\|_2^2 \\ &= \arg\min_{c\in\mathbb{R}} \left\| \frac{\langle \widetilde{X}_t^{(i-1)}, u_{t-1} + w_t X_t^{(i-1)} \rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2} \widetilde{X}_t^{(i-1)} - c\widetilde{X}_t^{(i-1)} \right\|_2^2 \\ &= \frac{\langle \widetilde{X}_t^{(i-1)}, u_{t-1} + w_t X_t^{(i-1)} \rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2}. \end{aligned}$$

Hence, a natural design of $q_t \in \mathscr{A}$ is to quantize c^*. Instead of using a deterministic quantizer as in [23, 38], we apply the stochastic quantizer in (5.6), that is

$$q_t := \mathscr{Q}_{\mathrm{StocQ}}(c^*) = \mathscr{Q}_{\mathrm{StocQ}}\left(\frac{\langle \widetilde{X}_t^{(i-1)}, u_{t-1} + w_t X_t^{(i-1)} \rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2} \right). \tag{5.14}$$

Putting everything together, the stochastic version of GPFQ, namely SPFQ in its basic form, can now be expressed as follows.

$$\begin{cases} u_0 = 0 \in \mathbb{R}^m, \\ q_t = \mathscr{Q}_{\mathrm{StocQ}}\left(\frac{\langle \widetilde{X}_t^{(i-1)}, u_{t-1} + w_t X_t^{(i-1)} \rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2} \right), \\ u_t = u_{t-1} + w_t X_t^{(i-1)} - q_t \widetilde{X}_t^{(i-1)} \end{cases} \tag{5.15}$$

where t iterates over $1, 2, \ldots, N_{i-1}$. In particular, the final error vector is

$$u_{N_{i-1}} = \sum_{j=1}^{N_{i-1}} w_j X_j^{(i-1)} - \sum_{j=1}^{N_{i-1}} q_j \widetilde{X}_j^{(i-1)} = X^{(i-1)} w - \widetilde{X}^{(i-1)} q \tag{5.16}$$

and our goal is to estimate $\|u_{N_{i-1}}\|_2$.

5.2.3 A two phases pipeline

An essential observation is that SPFQ in (5.15) can be equivalently decomposed into two phases.

Phase I: Given inputs $X^{(i-1)}$, $\widetilde{X}^{(i-1)}$ and neuron $w \in \mathbb{R}^{N_{i-1}}$ for the i-th layer, we first align the input data to the layer, by finding a real-valued vector $\widetilde{w} \in \mathbb{R}^{N_{i-1}}$ such that $\widetilde{X}^{(i-1)}\widetilde{w} \approx X^{(i-1)}w$. Similar to our discussion above (5.14), we adopt the same sequential selection strategy to obtain each $\widetilde{w}_t$ and deduce the following update rules.

$$\begin{cases} \hat{u}_0 = 0 \in \mathbb{R}^m, \\ \widetilde{w}_t = \frac{\langle \widetilde{X}_t^{(i-1)}, \hat{u}_{t-1} + w_t X_t^{(i-1)} \rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2}, \\ \hat{u}_t = \hat{u}_{t-1} + w_t X_t^{(i-1)} - \widetilde{w}_t \widetilde{X}_t^{(i-1)} \end{cases} \tag{5.17}$$

where $t = 1, 2 \ldots, N_{i-1}$. Note that the approximation error is given by

$$\hat{u}_{N_{i-1}} = X^{(i-1)}w - \widetilde{X}^{(i-1)}\widetilde{w}. \tag{5.18}$$

Phase II: After getting the new weights $\widetilde{w}$, we quantize $\widetilde{w}$ using SPFQ with input $\widetilde{X}^{(i-1)}$, i.e., finding $\widetilde{q} \in \mathscr{A}^{N_{i-1}}$ such that $\widetilde{X}^{(i-1)}\widetilde{q} \approx \widetilde{X}^{(i-1)}\widetilde{w}$. This process can be summarized as follows. For $t = 1, 2, \ldots, N_{i-1}$,

$$\begin{cases} \widetilde{u}_0 = 0 \in \mathbb{R}^m, \\ \widetilde{q}_t = \mathscr{Q}_{\text{StocQ}}\Big(\widetilde{w}_t + \frac{\langle \widetilde{X}_t^{(i-1)}, \widetilde{u}_{t-1} \rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2}\Big), \\ \widetilde{u}_t = \widetilde{u}_{t-1} + (\widetilde{w}_t - \widetilde{q}_t)\widetilde{X}_t^{(i-1)}. \end{cases} \tag{5.19}$$

Here, the quantization error is

$$\widetilde{u}_{N_{i-1}} = \widetilde{X}^{(i-1)}(\widetilde{w} - \widetilde{q}). \tag{5.20}$$

Proposition 5.2.2. *Given inputs $X^{(i-1)}$, $\widetilde{X}^{(i-1)}$ and any neuron $w \in \mathbb{R}^{N_{i-1}}$ for the i-th layer, the*

two phase formulation given by (5.17) *and* (5.19) *generate exactly same result as in* (5.15)*, that is,* $\widetilde{q} = q$.

Proof. We proceed by induction on the iteration index t. If $t = 1$, then (5.17), (5.19) and (5.15) imply that

$$\widetilde{q}_1 = \mathscr{Q}_{\text{StocQ}}(\widetilde{w}_1) = \mathscr{Q}_{\text{StocQ}}\Big(\frac{\langle \widetilde{X}_1^{(i-1)}, w_1 X_1^{(i-1)} \rangle}{\|\widetilde{X}_1^{(i-1)}\|_2^2}\Big) = q_1.$$

For $t \geq 2$, assume $\widetilde{q}_j = q_j$ for $1 \leq j \leq t-1$ and we aim to prove $\widetilde{q}_t = q_t$. Note that $\hat{u}_{t-1} = \sum_{j=1}^{t-1}(w_j X_j - \widetilde{w}_j \widetilde{X}_j)$ and $\widetilde{u}_{t-1} = \sum_{j=1}^{t-1}(\widetilde{w}_j \widetilde{X}_j - \widetilde{q}_j \widetilde{X}_j) = \sum_{j=1}^{t-1}(\widetilde{w}_j \widetilde{X}_j - q_j \widetilde{X}_j)$ by our induction hypothesis. It follows that $\hat{u}_{t-1} + \widetilde{u}_{t-1} = \sum_{j=1}^{t-1}(w_j X_j - q_j \widetilde{X}_j) = u_{t-1}$. Thus, we get

$$\widetilde{q}_t = \mathscr{Q}_{\text{StocQ}}\Big(\frac{\langle \widetilde{X}_t^{(i-1)}, \widetilde{u}_{t-1} + \hat{u}_{t-1} + w_t X_t^{(i-1)} \rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2}\Big) = \mathscr{Q}_{\text{StocQ}}\Big(\frac{\langle \widetilde{X}_t^{(i-1)}, u_{t-1} + w_t X_t^{(i-1)} \rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2}\Big) = q_t.$$

This establishes $\widetilde{q} = q$ and completes the proof. □

Based on Proposition 5.2.2, the quantization error (5.16) for SPFQ can be split into two parts:

$$u_{N_{i-1}} = X^{(i-1)} w - \widetilde{X}^{(i-1)} q = X^{(i-1)} w - \widetilde{X}^{(i-1)} \widetilde{w} + \widetilde{X}^{(i-1)}(\widetilde{w} - q) = \hat{u}_{N_{i-1}} + \widetilde{u}_{N_{i-1}}.$$

Here, the first error term $\hat{u}_{N_{i-1}}$ results from the data alignment in (5.17) to generate a new "virtual" neuron $\widetilde{w}$ and the second error term $\widetilde{u}_{N_{i-1}}$ is due to the quantization in (5.19). It follows that

$$\|u_{N_{i-1}}\|_2 = \|\hat{u}_{N_{i-1}} + \widetilde{u}_{N_{i-1}}\|_2 \leq \|\hat{u}_{N_{i-1}}\|_2 + \|\widetilde{u}_{N_{i-1}}\|_2. \tag{5.21}$$

Thus, we can bound the quantization error for SPFQ by controlling $\|\hat{u}_{N_{i-1}}\|_2$ and $\|\widetilde{u}_{N_{i-1}}\|_2$.

5.2.4 SPFQ Variants

The two-phase formulation of SPFQ provides a flexible framework that allows for the replacement of one or both phases with alternative algorithms. Here, our focus is on replacing

Algorithm 5: SPFQ with perfect data alignment

Input: An L-layer neural network Φ with weight matrices $W^{(i)} \in \mathbb{R}^{N_{i-1}\times N_i}$, input data $X \in \mathbb{R}^{m\times N_0}$

for $i = 1$ **to** L **do**

  Generate $X^{(i-1)} = \Phi^{(i-1)}(X) \in \mathbb{R}^{m\times N_{i-1}}$ and $\widetilde{X}^{(i-1)} = \widetilde{\Phi}^{(i-1)}(X) \in \mathbb{R}^{m\times N_{i-1}}$

  repeat For each column w of $W^{(i)}$

  Phase I: Find a solution $\widetilde{w}$ to (5.22)

  Phase II: Obtain the quantized neuron $\widetilde{q} \in \mathscr{A}^{N_{i-1}}$ via (5.19)

  until All columns of $W^{(i)}$ are quantized

  Obtain the quantized i-th layer weights $Q^{(i)} \in \mathscr{A}^{N_{i-1}\times N_i}$

Output: Quantized neural network $\widetilde{\Phi}$

the first, "data-alignment", phase to eliminate, or massively reduce, the error bound associated with this step. Indeed, by exploring alternative approaches, one can improve the error bounds of SPFQ, at the expense of increasing the computational complexity. Below, we present two such alternatives to Phase I.

In Section 5.3 we derive an error bound associated with the second phase of SPFQ, namely quantization, which is independent of the reconstructed neuron $\widetilde{w}$. Thus, to reduce the bound on $\|u_{N_{i-1}}\|_2$ in (5.21), we can eliminate $\|\hat{u}_{N_{i-1}}\|_2$ by simply choosing $\widetilde{w}$ with $\widetilde{X}^{(i-1)}\widetilde{w} = X^{(i-1)}w$. As this system of equations may admit infinitely many solutions, we opt for one with the minimal $\|\widetilde{w}\|_\infty$. This choice is motivated by the fact that smaller weights can be accommodated by smaller quantization alphabets, resulting in bit savings in practical applications. In other words, we replace Phase I with the optimization problem

$$\begin{aligned} \min_{\widetilde{w}\in\mathbb{R}^{N_{i-1}}} \quad & \|\widetilde{w}\|_\infty \\ \text{s.t.} \quad & \widetilde{X}^{(i-1)}\widetilde{w} = X^{(i-1)}w. \end{aligned} \tag{5.22}$$

It is not hard to see that (5.22) can be formulated as a linear program and solved via standard linear programming techniques [2]. Alternatively, powerful tools like Cadzow's method [4, 5] can also be used to solve linearly constrained infinity-norm optimization problems like (5.22).

Cadzow's method has computational complexity $O(m^2 N_{i-1})$, thus is a factor of m more expensive than our original approach but has the advantage of eliminating $\|\hat{u}_{N_i-1}\|_2$.

With this modification, one then proceeds with Phase II as before. Given a minimum ℓ_∞ solution $\widetilde{w}$ satisfying $\widetilde{X}^{(i-1)}\widetilde{w} = X^{(i-1)}w$, one can quantize it using (5.19) and obtain $\widetilde{q} \in \mathscr{A}^{N_{i-1}}$. In this case, $\widetilde{q}$ may not be equal to q in (5.15) and the quantization error becomes

$$X^{(i-1)}w - \widetilde{X}^{(i-1)}\widetilde{q} = \widetilde{X}^{(i-1)}(\widetilde{w} - \widetilde{q}) = \widetilde{u}_{N_{i-1}} \tag{5.23}$$

where only Phase II is involved. We summarize this version of SPFQ in Algorithm 5.

The second approach we present herein aims to reduce the computational complexity associated with (5.22). To that end, we generalize the data alignment process in (5.17) as follows. Let $r \in \mathbb{Z}^+$ and $w \in \mathbb{R}^{N_{i-1}}$. For $t = 1, 2, \ldots, N_{i-1}$, we perform (5.17) as before. Now however, for $t = N_{i-1}+1, N_{i-1}+2, \ldots, rN_{i-1}$, we run

$$\begin{cases} \hat{v}_{t-1} = \hat{u}_{t-1} - w_t X_t^{(i-1)} + \widetilde{w}_t \widetilde{X}_t^{(i-1)}, \\ \widetilde{w}_t = \frac{\langle \widetilde{X}_t^{(i-1)}, \hat{v}_{t-1} + w_t X_t^{(i-1)} \rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2}, \\ \hat{u}_t = \hat{v}_{t-1} + w_t X_t^{(i-1)} - \widetilde{w}_t \widetilde{X}_t^{(i-1)} \end{cases} \tag{5.24}$$

Here, we use modulo N_{i-1} indexing for (the subscripts of) $w, \widetilde{w}, X^{(i-1)}$, and $\widetilde{X}^{(i-1)}$. We call the combination of (5.17) and (5.24) the r-th order data alignment procedure, which costs $O(rmN_{i-1})$ operations. Applying (5.19) to the output $\widetilde{w}$ as before, the quantization error consists of two parts:

$$X^{(i-1)}w - \widetilde{X}^{(i-1)}\widetilde{q} = X^{(i-1)}w - \widetilde{X}^{(i-1)}\widetilde{w} + \widetilde{X}^{(i-1)}(\widetilde{w} - \widetilde{q}) = \hat{u}_{rN_{i-1}} + \widetilde{u}_{N_{i-1}}. \tag{5.25}$$

This version of SPFQ with order r is summarized in Algorithm 6. In Section 5.3, we prove that the data alignment error $\hat{u}_{rN_{i-1}} = X^{(i-1)}w - \widetilde{X}^{(i-1)}\widetilde{w}$ decays exponentially in order r.

Algorithm 6: SPFQ with approximated data alignment

Input: An L-layer neural network Φ with weight matrices $W^{(i)} \in \mathbb{R}^{N_{i-1}\times N_i}$, input data $X \in \mathbb{R}^{m\times N_0}$, order $r \in \mathbb{Z}^+$

for $i = 1$ **to** L **do**

  Generate $X^{(i-1)} = \Phi^{(i-1)}(X) \in \mathbb{R}^{m\times N_{i-1}}$ and $\widetilde{X}^{(i-1)} = \widetilde{\Phi}^{(i-1)}(X) \in \mathbb{R}^{m\times N_{i-1}}$

  repeat For each column of $W^{(i)}$

  Phase I: Pick a column (neuron) $w \in \mathbb{R}^{N_{i-1}}$ of $W^{(i)}$ and get $\widetilde{w}$ using the r-th order data alignment in (5.17) and (5.24)

  Phase II: Quantize $\widetilde{w}$ via (5.19)

  until All columns of $W^{(i)}$ are quantized

  Obtain quantized i-th layer $Q^{(i)} \in \mathscr{A}^{N_{i-1}\times N_i}$

Output: Quantized neural network $\widetilde{\Phi}$

5.3 Error Bounds for SPFQ with Infinite Alphabets

We can now begin analyzing the errors associated with the above variants of SPFQ. On the one hand, in Algorithm 5, since data is perfectly aligned by solving (5.22), we only have to bound the quantization error $\widetilde{u}_{N_{i-1}}$ generated by procedure (5.19). On the other hand, Algorithm 6 has a faster implementation provided $r < m$, but introduces an extra error $\hat{u}_{rN_{i-1}}$ arising from the r-th order data alignment. Thus, to control the error bounds for this version of SPFQ, we first bound $\widetilde{u}_{N_{i-1}}$ and $\hat{u}_{rN_{i-1}}$ appearing in (5.23) and (5.25).

Lemma 5.3.1 (Quantization error)**.** *Assuming that the first $i-1$ layers have been quantized, let $X^{(i-1)}$, $\widetilde{X}^{(i-1)}$ be as in* (5.11) *and $w \in \mathbb{R}^{N_{i-1}}$ be the weights associated with a neuron in the i-th layer, i.e. a column of $W^{(i)} \in \mathbb{R}^{N_{i-1}\times N_i}$. Suppose $\widetilde{w}$ is either the solution of* (5.22) *or the output of* (5.24)*. Quantize $\widetilde{w}$ using* (5.19) *with alphabets $\mathscr{A} = \mathscr{A}_\infty^\delta$ as in* (5.4)*. Then, for any $p \in \mathbb{N}$,*

$$\|\widetilde{u}_{N_{i-1}}\|_2 \le \delta\sqrt{2\pi pm\log N_{i-1}}\max_{1\le j\le N_{i-1}}\|\widetilde{X}_j^{(i-1)}\|_2 \tag{5.26}$$

holds with probability at least $1-\frac{\sqrt{2}m}{N_{i-1}^p}$.

Proof. We first show that

$$\widetilde{u}_t \le_{\mathrm{cx}} \mathscr{N}(0,\Sigma_t) \tag{5.27}$$

holds for all $1 \le t \le N_{i-1}$, where Σ_t is defined recursively as follows

$$\Sigma_t := P_{\widetilde{X}_t^{(i-1)\perp}} \Sigma_{t-1} P_{\widetilde{X}_t^{(i-1)\perp}} + \frac{\pi\delta^2}{2} \widetilde{X}_t^{(i-1)} \widetilde{X}_t^{(i-1)\top} \quad \text{with} \quad \Sigma_0 := 0.$$

At the t-th step of quantizing $\widetilde{w}$, by (5.19), we have $\widetilde{u}_t = \widetilde{u}_{t-1} + (\widetilde{w}_t - \widetilde{q}_t)\widetilde{X}_t^{(i-1)}$. Define

$$h_t := \widetilde{u}_{t-1} + \widetilde{w}_t \widetilde{X}_t^{(i-1)} \quad \text{and} \quad v_t := \frac{\langle \widetilde{X}_t^{(i-1)}, h_t \rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2}. \tag{5.28}$$

It follows that

$$\widetilde{u}_t = h_t - \widetilde{q}_t \widetilde{X}_t^{(i-1)} \tag{5.29}$$

and (5.19) implies

$$\widetilde{q}_t = \mathscr{Q}_{\mathrm{StocQ}}\left(\frac{\langle \widetilde{X}_t^{(i-1)}, h_t \rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2} \right) = \mathscr{Q}_{\mathrm{StocQ}}(v_t). \tag{5.30}$$

Since $\mathscr{A} = \mathscr{A}_\infty^\delta$, $\mathbb{E}\mathscr{Q}_{\mathrm{StocQ}}(z) = z$ for all $z \in \mathbb{R}$. Moreover, conditioning on $\widetilde{u}_{t-1}$ in (5.28), h_t and v_t are fixed and thus one can get

$$\mathbb{E}(\mathscr{Q}_{\mathrm{StocQ}}(v_t) | \widetilde{u}_{t-1}) = v_t \tag{5.31}$$

and

$$\begin{aligned}
\mathbb{E}(\widetilde{u}_t|\widetilde{u}_{t-1}) &= \mathbb{E}(h_t - \widetilde{q}_t\widetilde{X}_t^{(i-1)}|\widetilde{u}_{t-1}) \\
&= h_t - \widetilde{X}_t^{(i-1)}\mathbb{E}(\widetilde{q}_t|\widetilde{u}_{t-1}) \\
&= h_t - \widetilde{X}_t^{(i-1)}\mathbb{E}(\mathscr{Q}_{\mathrm{StocQ}}(v_t)|\widetilde{u}_{t-1}) \\
&= h_t - v_t\widetilde{X}_t^{(i-1)} \\
&= h_t - \frac{\langle \widetilde{X}_t^{(i-1)}, h_t\rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2}\widetilde{X}_t^{(i-1)} \\
&= \left(I - \frac{\widetilde{X}_t^{(i-1)}\widetilde{X}_t^{(i-1)\top}}{\|\widetilde{X}_t^{(i-1)}\|_2^2}\right)h_t \\
&= P_{\widetilde{X}_t^{(i-1)\perp}}(h_t).
\end{aligned}$$

The identity above indicates that the approximation error $\widetilde{u}_t$ can be split into two parts: its conditional mean $P_{\widetilde{X}_t^{(i-1)\perp}}(h_t)$ and a random perturbation. Specifically, applying (5.29) and (5.7), we obtain

$$\widetilde{u}_t = P_{\widetilde{X}_t^{(i-1)\perp}}(h_t) + P_{\widetilde{X}_t^{(i-1)}}(h_t) - \widetilde{q}_t\widetilde{X}_t^{(i-1)} = P_{\widetilde{X}_t^{(i-1)\perp}}(h_t) + R_t\widetilde{X}_t^{(i-1)} \tag{5.32}$$

where

$$R_t := v_t - \widetilde{q}_t.$$

Further, combining (5.30) and (5.31), we have

$$\mathbb{E}(R_t|\widetilde{u}_{t-1}) = v_t - \mathbb{E}(\widetilde{q}_t|\widetilde{u}_{t-1}) = v_t - \mathbb{E}(\mathscr{Q}_{\mathrm{StocQ}}(v_t)|\widetilde{u}_{t-1}) = 0$$

and $|R_t| = |v_t - \mathscr{Q}_{\mathrm{StocQ}}(v_t)| \le \delta$. Lemma 5.6.5 yields that, conditioning on $\widetilde{u}_{t-1}$,

$$R_t \le_{\mathrm{cx}} \mathscr{N}\left(0, \frac{\pi\delta^2}{2}\right). \tag{5.33}$$

Now, we are ready to prove (5.27) by induction on t. When $t = 1$, we have $h_1 = \widetilde{w}_1\widetilde{X}_1^{(i-1)}$.

We can deduce from (5.32) and (5.33) that $\widetilde{u}_1 = P_{\widetilde{X}_1^{(i-1)\perp}}(\widetilde{w}_1 \widetilde{X}_1^{(i-1)}) + R_1 \widetilde{X}_1^{(i-1)} = R_1 \widetilde{X}_1^{(i-1)}$ with $R_1 \leq_{\text{cx}} \mathcal{N}\big(0, \frac{\pi\delta^2}{2}\big)$. Applying Lemma 5.6.3, we obtain $\widetilde{u}_1 \leq_{\text{cx}} \mathcal{N}(0, \Sigma_1)$. Next, assume that (5.27) holds for $t-1$ with $t \geq 2$. By the induction hypothesis, we have $\widetilde{u}_{t-1} \leq_{\text{cx}} \mathcal{N}(0, \Sigma_{t-1})$. Using Lemma 5.6.3 again, we get

$$\begin{aligned} P_{\widetilde{X}_t^{(i-1)\perp}}(h_t) &= P_{\widetilde{X}_t^{(i-1)\perp}}(\widetilde{u}_{t-1} + \widetilde{w}_t \widetilde{X}_t^{(i-1)}) \\ &\leq_{\text{cx}} \mathcal{N}\Big(P_{\widetilde{X}_t^{(i-1)\perp}}(\widetilde{w}_t \widetilde{X}_t^{(i-1)}), P_{\widetilde{X}_t^{(i-1)\perp}} \Sigma_{t-1} P_{\widetilde{X}_t^{(i-1)\perp}}\Big) \\ &= \mathcal{N}\Big(0, P_{\widetilde{X}_t^{(i-1)\perp}} \Sigma_{t-1} P_{\widetilde{X}_t^{(i-1)\perp}}\Big). \end{aligned}$$

Additionally, conditioning on $\widetilde{u}_{t-1}$, (5.33) implies

$$R_t \widetilde{X}_t^{(i-1)} \leq_{\text{cx}} \mathcal{N}\Big(0, \frac{\pi\delta^2}{2} \widetilde{X}_t^{(i-1)} \widetilde{X}_t^{(i-1)\top}\Big).$$

Then we apply Lemma 5.6.4 to (5.32) by taking

$$X = P_{\widetilde{X}_t^{(i-1)\perp}}(h_t), Y = \widetilde{u}_t, W = \mathcal{N}\Big(0, P_{\widetilde{X}_t^{(i-1)\perp}} \Sigma_{t-1} P_{\widetilde{X}_t^{(i-1)\perp}}\Big), Z = \mathcal{N}\Big(0, \frac{\pi\delta^2}{2} \widetilde{X}_t^{(i-1)} \widetilde{X}_t^{(i-1)\top}\Big).$$

It follows that

$$\begin{aligned} \widetilde{u}_t &\leq_{\text{cx}} W + Z \\ &= \mathcal{N}\Big(0, P_{\widetilde{X}_t^{(i-1)\perp}} \Sigma_{t-1} P_{\widetilde{X}_t^{(i-1)\perp}} + \frac{\pi\delta^2}{2} \widetilde{X}_t^{(i-1)} \widetilde{X}_t^{(i-1)\top}\Big) \\ &= \mathcal{N}(0, \Sigma_t). \end{aligned}$$

Here, we used the independence of W and Z, and the definition of Σ_t. This establishes inequality (5.27) showing that $\widetilde{u}_t$ is dominated by $\mathcal{N}(0, \Sigma_t)$ in the convex order, where Σ_t is defined recursively using orthogonal projections. So it remains to control the covariance matrix Σ_t.

Recall that Σ_t is defined as follows.

$$\Sigma_t = P_{\widetilde{X}_t^{(i-1)\perp}} \Sigma_{t-1} P_{\widetilde{X}_t^{(i-1)\perp}} + \frac{\pi\delta^2}{2} \widetilde{X}_t^{(i-1)} \widetilde{X}_t^{(i-1)\top} \quad \text{with} \quad \Sigma_0 = 0.$$

Then we apply Lemma 5.7.1 with $M_t = \Sigma_t$, $z_t = \widetilde{X}_t^{(i-1)}$, and $\alpha = \frac{\pi\delta^2}{2}$, and conclude that $\Sigma_t \preceq \sigma_t^2 I$ with $\sigma_t^2 = \frac{\pi\delta^2}{2} \max_{1\le j\le t} \|\widetilde{X}_j^{(i-1)}\|_2^2$. Note that $\widetilde{u}_t \le_{\mathrm{cx}} \mathcal{N}(0,\Sigma_t)$ and, by Lemma 5.6.2, we have $\mathcal{N}(0,\Sigma_t) \le_{\mathrm{cx}} \mathcal{N}(0,\sigma_t^2 I)$. Then we deduce from the transitivity of $\le_{\mathrm{cx}}$ that $\widetilde{u}_t \le_{\mathrm{cx}} \mathcal{N}(0,\sigma_t^2 I)$. It follows from Lemma 5.7.2 that, for $\gamma \in (0,1]$ and $1 \le t \le N_{i-1}$,

$$\mathrm{P}\left(\|\widetilde{u}_t\|_\infty \le 2\sigma_t \sqrt{\log(\sqrt{2}m/\gamma)} \right) \ge 1-\gamma.$$

Picking $\gamma = \sqrt{2}mN_{i-1}^{-p}$ and $t = N_{i-1}$,

$$\|\widetilde{u}_{N_{i-1}}\|_2 \le \sqrt{m}\|\widetilde{u}_{N_{i-1}}\|_\infty \le 2\sigma_{N_{i-1}} \sqrt{pm\log N_{i-1}} = \delta\sqrt{2\pi pm\log N_{i-1}} \max_{1\le j\le N_{i-1}} \|\widetilde{X}_j^{(i-1)}\|_2$$

holds with probability exceeding $1-\sqrt{2}mN_{i-1}^{-p}$. □

Next, we deduce a closed-form expression of $\hat{u}_{rN_{i-1}}$ showing that $\|\hat{u}_{rN_{i-1}}\|_2$ decays polynomially with respect to r.

Lemma 5.3.2 (Data alignment error)**.** *Assuming that the first $i-1$ layers have been quantized, let $X^{(i-1)}$, $\widetilde{X}^{(i-1)}$ be as in* (5.11) *and let $w \in \mathbb{R}^{N_{i-1}}$ be a neuron in the i-th layer, i.e. a column of $W^{(i)} \in \mathbb{R}^{N_{i-1}\times N_i}$. Applying the r-th order data alignment procedure in* (5.17) *and* (5.24)*, we have*

$$\hat{u}_{N_{i-1}} = \sum_{j=1}^{N_{i-1}} w_j P_{\widetilde{X}_{N_{i-1}}^{(i-1)\perp}} \dots P_{\widetilde{X}_{j+1}^{(i-1)\perp}} P_{\widetilde{X}_j^{(i-1)\perp}} (X_j^{(i-1)}) \tag{5.34}$$

and

$$\hat{u}_{rN_{i-1}} = (P^{(i-1)})^{r-1} \hat{u}_{N_{i-1}} \tag{5.35}$$

where $P^{(i-1)} := P_{\widetilde{X}_{N_{i-1}}^{(i-1)\perp}} \dots P_{\widetilde{X}_2^{(i-1)\perp}} P_{\widetilde{X}_1^{(i-1)\perp}}$.

Proof. We first prove the following identity by induction on t.

$$\hat{u}_t = \sum_{j=1}^{t} w_j P_{\widetilde{X}_t^{(i-1)\perp}} \dots P_{\widetilde{X}_{j+1}^{(i-1)\perp}} P_{\widetilde{X}_j^{(i-1)\perp}}(X_j^{(i-1)}), \quad 1 \le t \le N_{i-1}. \tag{5.36}$$

By (5.17), the case $t = 1$ is straightforward, since we have

$$\begin{aligned}
\hat{u}_1 &= w_1 X_1^{(i-1)} - \widetilde{w}_1 \widetilde{X}_1^{(i-1)} \\
&= w_1 X_1^{(i-1)} - \frac{\langle \widetilde{X}_1^{(i-1)}, w_1 X_1^{(i-1)} \rangle}{\|\widetilde{X}_1^{(i-1)}\|_2^2} \widetilde{X}_1^{(i-1)} \\
&= w_1 X_1^{(i-1)} - P_{\widetilde{X}_1^{(i-1)}}(w_1 X_1^{(i-1)}) \\
&= w_1 P_{\widetilde{X}_1^{(i-1)\perp}}(X_1^{(i-1)})
\end{aligned}$$

where we apply the properties of orthogonal projections in (5.7) and (5.8). For $2 \le t \le N_{i-1}$, assume that (5.36) holds for $t-1$. Then, by (5.17), one gets

$$\begin{aligned}
\hat{u}_t &= \hat{u}_{t-1} + w_t X_t^{(i-1)} - \widetilde{w}_t \widetilde{X}_t^{(i-1)} \\
&= \hat{u}_{t-1} + w_t X_t^{(i-1)} - \frac{\langle \widetilde{X}_t^{(i-1)}, \hat{u}_{t-1} + w_t X_t^{(i-1)} \rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2} \widetilde{X}_t^{(i-1)} \\
&= \hat{u}_{t-1} + w_t X_t^{(i-1)} - P_{\widetilde{X}_t^{(i-1)}}(\hat{u}_{t-1} + w_t X_t^{(i-1)}) \\
&= P_{\widetilde{X}_t^{(i-1)\perp}}(\hat{u}_{t-1} + w_t X_t^{(i-1)}).
\end{aligned}$$

Applying the induction hypothesis, we obtain

$$\begin{aligned}
\hat{u}_t &= P_{\widetilde{X}_t^{(i-1)\perp}}(\hat{u}_{t-1}) + w_t P_{\widetilde{X}_t^{(i-1)\perp}}(X_t^{(i-1)}) \\
&= \sum_{j=1}^{t-1} w_j P_{\widetilde{X}_t^{(i-1)\perp}} \dots P_{\widetilde{X}_{j+1}^{(i-1)\perp}} P_{\widetilde{X}_j^{(i-1)\perp}}(X_j^{(i-1)}) + w_t P_{\widetilde{X}_t^{(i-1)\perp}}(X_t^{(i-1)}) \\
&= \sum_{j=1}^{t} w_j P_{\widetilde{X}_t^{(i-1)\perp}} \dots P_{\widetilde{X}_{j+1}^{(i-1)\perp}} P_{\widetilde{X}_j^{(i-1)\perp}}(X_j^{(i-1)}).
\end{aligned}$$

This completes the proof of (5.36). In particular, if $t = N_{i-1}$, then we obtain (5.34).

Next, we consider $\hat{u}_t$ when $t > N_{i-1}$. Plugging $t = N_{i-1}+1$ into (5.24), and recalling that our indices (except for $\hat{u}$) are modulo N_{i-1}, we have

$$\hat{u}_{N_{i-1}+1} = \hat{u}_{N_{i-1}} + \widetilde{w}_1 \widetilde{X}_1^{(i-1)} - \frac{\langle \widetilde{X}_1^{(i-1)}, \hat{u}_{N_{i-1}} + \widetilde{w}_1 \widetilde{X}_1^{(i-1)} \rangle}{\|\widetilde{X}_1^{(i-1)}\|_2^2} \widetilde{X}_1^{(i-1)} = P_{\widetilde{X}_1^{(i-1)\perp}}(\hat{u}_{N_{i-1}}).$$

Similarly, one can show that $\hat{u}_{N_{i-1}+2} = P_{\widetilde{X}_2^{(i-1)\perp}}(\hat{u}_{N_{i-1}+1}) = P_{\widetilde{X}_2^{(i-1)\perp}} P_{\widetilde{X}_1^{(i-1)\perp}} \hat{u}_{N_{i-1}}$. Repeating this argument for all $N_{i-1} < t \le rN_{i-1}$, we can derive (5.35). □

Combining Lemma 5.3.1 and Lemma 5.3.2, we can derive a recursive relation between the error in the current layer and that of the previous layer.

Theorem 5.3.3. *Let Φ be an L-layer neural network as in* (5.3) *where the activation function is $\varphi^{(i)}(x) = \rho(x) := \max\{0, x\}$ for $1 \le i \le L$. Let $\mathscr{A} = \mathscr{A}_\infty^\delta$ be as in* (5.4) *and $p \in \mathbb{N}$.*

(a) If we quantize Φ using Algorithm 5, then, for each $2 \le i \le L$,

$$\max_{1\le j\le N_i} \|X^{(i-1)} W_j^{(i)} - \widetilde{X}^{(i-1)} Q_j^{(i)}\|_2 \le \delta \sqrt{2\pi pm \log N_{i-1}} \max_{1\le j\le N_{i-1}} \|X_j^{(i-1)}\|_2$$
$$+ \delta \sqrt{2\pi pm \log N_{i-1}} \max_{1\le j\le N_{i-1}} \|X^{(i-2)} W_j^{(i-1)} - \widetilde{X}^{(i-2)} Q_j^{(i-1)}\|_2.$$

holds with probability at least $1 - \frac{\sqrt{2}mN_i}{N_{i-1}^p}$.

(b) If we quantize Φ using Algorithm 6, then, for each $2 \le i \le L$,

$$\max_{1\le j\le N_i} \|X^{(i-1)} W_j^{(i)} - \widetilde{X}^{(i-1)} Q_j^{(i)}\|_2 \le \delta \sqrt{2\pi pm \log N_{i-1}} \max_{1\le j\le N_{i-1}} \|X_j^{(i-1)}\|_2$$
$$+ \left(N_{i-1} \|W^{(i)}\|_{\max} \|P^{(i-1)}\|_2^{r-1} + \delta \sqrt{2\pi pm \log N_{i-1}} \right) \max_{1\le j\le N_{i-1}} \|X^{(i-2)} W_j^{(i-1)} - \widetilde{X}^{(i-2)} Q_j^{(i-1)}\|_2$$

holds with probability exceeding $1 - \frac{\sqrt{2}mN_i}{N_{i-1}^p}$. Here, $P^{(i-1)}$ is defined in Lemma 5.3.2.

Proof. (a) Note that, for each $1 \le j \le N_i$, the j-th columns $W_j^{(i)}$ and $Q_j^{(i)}$ represent a neuron and

its quantized version respectively. Applying (5.23) and (5.26), we obtain

$$\mathrm{P}\Big(\|X^{(i-1)}W_j^{(i)} - \widetilde{X}^{(i-1)}Q_j^{(i)}\|_2 \le \delta\sqrt{2\pi pm\log N_{i-1}}\max_{1\le j\le N_{i-1}}\|\widetilde{X}_j^{(i-1)}\|_2\Big) \ge 1 - \frac{\sqrt{2}m}{N_{i-1}^p}.$$

Taking a union bound over all j,

$$\max_{1\le j\le N_i}\|X^{(i-1)}W_j^{(i)} - \widetilde{X}^{(i-1)}Q_j^{(i)}\|_2 \le \delta\sqrt{2\pi pm\log N_{i-1}}\max_{1\le j\le N_{i-1}}\|\widetilde{X}_j^{(i-1)}\|_2$$

holds with probability at least $1 - \frac{\sqrt{2}mN_i}{N_{i-1}^p}$. By the triangle inequality, we have

$$\begin{aligned}\max_{1\le j\le N_{i-1}}\|\widetilde{X}_j^{(i-1)}\|_2 &\le \max_{1\le j\le N_{i-1}}\|X_j^{(i-1)}\|_2 + \max_{1\le j\le N_{i-1}}\|X_j^{(i-1)} - \widetilde{X}_j^{(i-1)}\|_2 \\ &= \max_{1\le j\le N_{i-1}}\|X_j^{(i-1)}\|_2 + \max_{1\le j\le N_{i-1}}\|\rho(X^{(i-2)}W_j^{(i-1)}) - \rho(\widetilde{X}^{(i-2)}Q_j^{(i-1)})\|_2 \\ &\le \max_{1\le j\le N_{i-1}}\|X_j^{(i-1)}\|_2 + \max_{1\le j\le N_{i-1}}\|X^{(i-2)}W_j^{(i-1)} - \widetilde{X}^{(i-2)}Q_j^{(i-1)}\|_2 \qquad (5.37)\end{aligned}$$

It follows that, with probability at least $1 - \frac{\sqrt{2}mN_i}{N_{i-1}^p}$,

$$\begin{aligned}&\max_{1\le j\le N_i}\|X^{(i-1)}W_j^{(i)} - \widetilde{X}^{(i-1)}Q_j^{(i)}\|_2 \le \delta\sqrt{2\pi pm\log N_{i-1}}\max_{1\le j\le N_{i-1}}\|X_j^{(i-1)}\|_2 \\ &+ \delta\sqrt{2\pi pm\log N_{i-1}}\max_{1\le j\le N_{i-1}}\|X^{(i-2)}W_j^{(i-1)} - \widetilde{X}^{(i-2)}Q_j^{(i-1)}\|_2.\end{aligned}$$

(b) Applying Lemma 5.3.2 with $w = W_j^{(i)}$ and using the fact that $\|P\|_2 \le 1$ for any orthogonal

projection P, we have

$$\begin{aligned}
\|\hat{u}_{N_{i-1}}\|_2 &= \Big\| \sum_{k=1}^{N_{i-1}} W_{kj}^{(i)} P_{\widetilde{X}_{N_{i-1}}^{(i-1)\perp}} \dots P_{\widetilde{X}_{k+1}^{(i-1)\perp}} P_{\widetilde{X}_k^{(i-1)\perp}} (X_k^{(i-1)}) \Big\|_2 \\
&\le \sum_{k=1}^{N_{i-1}} |W_{kj}^{(i)}| \Big\| P_{\widetilde{X}_k^{(i-1)\perp}} (X_k^{(i-1)}) \Big\|_2 \\
&= \sum_{k=1}^{N_{i-1}} |W_{kj}^{(i)}| \Big\| P_{\widetilde{X}_k^{(i-1)\perp}} (X_k^{(i-1)} - \widetilde{X}_k^{(i-1)}) \Big\|_2 \\
&\le N_{i-1} \|W_j^{(i)}\|_\infty \max_{1\le j\le N_{i-1}} \|X_j^{(i-1)} - \widetilde{X}_j^{(i-1)}\|_2 \\
&= N_{i-1} \|W_j^{(i)}\|_\infty \max_{1\le j\le N_{i-1}} \|\rho(X^{(i-2)} W_j^{(i-1)}) - \rho(\widetilde{X}^{(i-2)} Q_j^{(i-1)})\|_2 \\
&\le N_{i-1} \|W^{(i)}\|_{\max} \max_{1\le j\le N_{i-1}} \|X^{(i-2)} W_j^{(i-1)} - \widetilde{X}^{(i-2)} Q_j^{(i-1)}\|_2. \qquad (5.38)
\end{aligned}$$

Then it follows from (5.25), (5.26), (5.37), and (5.38) that

$$\begin{aligned}
&\|X^{(i-1)} W_j^{(i)} - \widetilde{X}^{(i-1)} Q_j^{(i)}\|_2 \\
&\le \|\hat{u}_{rN_{i-1}}\|_2 + \|\widetilde{u}_{N_{i-1}}\|_2 \\
&\le \|P^{(i-1)}\|_2^{r-1} \|\hat{u}_{N_{i-1}}\|_2 + \delta \sqrt{2\pi pm \log N_{i-1}} \max_{1\le j\le N_{i-1}} \|\widetilde{X}_j^{(i-1)}\|_2 \\
&\le N_{i-1} \|W^{(i)}\|_{\max} \|P^{(i-1)}\|_2^{r-1} \max_{1\le j\le N_{i-1}} \|X^{(i-2)} W_j^{(i-1)} - \widetilde{X}^{(i-2)} Q_j^{(i-1)}\|_2 + \delta \sqrt{2\pi pm \log N_{i-1}} \\
&\times \Big(\max_{1\le j\le N_{i-1}} \|X_j^{(i-1)}\|_2 + \max_{1\le j\le N_{i-1}} \|X^{(i-2)} W_j^{(i-1)} - \widetilde{X}^{(i-2)} Q_j^{(i-1)}\|_2 \Big)
\end{aligned}$$

holds with probability at least $1 - \sqrt{2} m N_{i-1}^{-p}$. By a union bound over all j, we obtain that

$$\begin{aligned}
&\max_{1\le j\le N_i} \|X^{(i-1)} W_j^{(i)} - \widetilde{X}^{(i-1)} Q_j^{(i)}\|_2 \le \delta \sqrt{2\pi pm \log N_{i-1}} \max_{1\le j\le N_{i-1}} \|X_j^{(i-1)}\|_2 \\
&+ \Big(N_{i-1} \|W^{(i)}\|_{\max} \|P^{(i-1)}\|_2^{r-1} + \delta \sqrt{2\pi pm \log N_{i-1}} \Big) \max_{1\le j\le N_{i-1}} \|X^{(i-2)} W_j^{(i-1)} - \widetilde{X}^{(i-2)} Q_j^{(i-1)}\|_2
\end{aligned}$$

holds with probability exceeding $1 - \frac{\sqrt{2} m N_i}{N_{i-1}^p}$. □

Applying Theorem 5.3.3 inductively for all layers, one can obtain an error bound for quantizing the whole neural network.

Corollary 5.3.4. *Let* Φ *be an L-layer neural network as in* (5.3) *where the activation function is* $\varphi^{(i)}(x) = \rho(x) := \max\{0,x\}$ *for* $1 \le i \le L$. *Let* $\mathscr{A} = \mathscr{A}_\infty^\delta$ *be as in* (5.4) *and* $p \in \mathbb{N}$.

(a) *If we quantize* Φ *using Algorithm 5, then*

$$\max_{1\le j\le N_L} \|\Phi(X)_j - \widetilde{\Phi}(X)_j\|_2 \le \sum_{i=0}^{L-1} (2\pi pm\delta^2)^{\frac{L-i}{2}} \Big(\prod_{k=i}^{L-1} \log N_k\Big)^{\frac{1}{2}} \max_{1\le j\le N_i} \|X_j^{(i)}\|_2 \tag{5.39}$$

holds with probability at least $1 - \sum_{i=1}^{L} \frac{\sqrt{2}mN_i}{N_{i-1}^p}$.

(b) *If we quantize* Φ *using Algorithm 6, then*

$$\begin{aligned}&\max_{1\le j\le N_L} \|\Phi(X)_j - \widetilde{\Phi}(X)_j\|_2 \le \\ &\sum_{i=0}^{L-1} \delta\sqrt{2\pi pm\log N_i} \max_{1\le j\le N_i} \|X_j^{(i)}\|_2 \prod_{k=i+1}^{L-1} \Big(N_k \|W^{(k+1)}\|_{\max} \|P^{(k)}\|_2^{r-1} + \delta\sqrt{2\pi pm\log N_k}\Big)\end{aligned} \tag{5.40}$$

holds with probability at least $1 - \sum_{i=1}^{L} \frac{\sqrt{2}mN_i}{N_{i-1}^p}$. *Here,* $P^{(k)} = P_{\widetilde{X}_{N_k}^{(k)\perp}} \dots P_{\widetilde{X}_2^{(k)\perp}} P_{\widetilde{X}_1^{(k)\perp}}$ *is defined in Lemma 5.3.2.*

Proof. (a) For $1 \le j \le N_L$, by (5.11), we have

$$\Phi(X)_j = X_j^{(L)} = \rho(X^{(L-1)} W_j^{(L)}) \quad \text{and} \quad \widetilde{\Phi}(X)_j = \widetilde{X}_j^{(L)} = \rho(\widetilde{X}^{(L-1)} Q_j^{(L)})$$

where $W_j^{(L)}$ and $Q_j^{(L)}$ are the j-th neuron in the L-th layer and its quantized version respectively.

It follows from part (a) of Theorem 5.3.3 with $i = L$ that

$$\max_{1\le j\le N_L} \|\Phi(X)_j - \widetilde{\Phi}(X)_j\|_2 = \max_{1\le j\le N_L} \|\rho(X^{(L-1)}W_j^{(L)}) - \rho(\widetilde{X}^{(L-1)}Q_j^{(L)})\|_2$$
$$\le \max_{1\le j\le N_L} \|X^{(L-1)}W_j^{(L)} - \widetilde{X}^{(L-1)}Q_j^{(L)}\|_2$$
$$\le \delta\sqrt{2\pi pm\log N_{L-1}} \max_{1\le j\le N_{L-1}} \|X_j^{(L-1)}\|_2$$
$$+ \delta\sqrt{2\pi pm\log N_{L-1}} \max_{1\le j\le N_{L-1}} \|X^{(L-2)}W_j^{(L-1)} - \widetilde{X}^{(L-2)}Q_j^{(L-1)}\|_2.$$

holds with probability at least $1 - \frac{\sqrt{2}mN_L}{N_{L-1}^p}$. Moreover, by applying part (a) of Theorem 5.3.3 with $i = L-1$ to the result above, we obtain that

$$\max_{1\le j\le N_L} \|\Phi(X)_j - \widetilde{\Phi}(X)_j\|_2 \le \delta\sqrt{2\pi pm\log N_{L-1}} \max_{1\le j\le N_{L-1}} \|X_j^{(L-1)}\|_2 + 2\pi pm\delta^2$$
$$\times \sqrt{\log N_{L-1}\log N_{L-2}}\Big(\max_{1\le j\le N_{L-2}} \|X_j^{(L-2)}\|_2 + \max_{1\le j\le N_{i-1}} \|X^{(i-2)}W_j^{(i-1)} - \widetilde{X}^{(i-2)}Q_j^{(i-1)}\|_2\Big)$$

holds with probability at least $1 - \frac{\sqrt{2}mN_L}{N_{L-1}^p} - \frac{\sqrt{2}mN_{L-1}}{N_{L-2}^p}$. Repeating this argument inductively for $i = L-2, L-3, \ldots, 1$, one can derive

$$\max_{1\le j\le N_L} \|\Phi(X)_j - \widetilde{\Phi}(X)_j\|_2 \le \sum_{i=0}^{L-1} (2\pi pm\delta^2)^{\frac{L-i}{2}} \Big(\prod_{k=i}^{L-1} \log N_k\Big)^{\frac{1}{2}} \max_{1\le j\le N_i} \|X_j^{(i)}\|_2$$

with probability at least $1 - \sum_{i=1}^{L} \frac{\sqrt{2}mN_i}{N_{i-1}^p}$.

(b) The proof of (5.40) is similar to the one we had in part (a) except that we need to use part (b)

of Theorem 5.3.3 this time. Indeed, for the case of $i = L$,

$$\begin{aligned}
&\max_{1\le j\le N_L} \|\Phi(X)_j - \widetilde{\Phi}(X)_j\|_2 = \max_{1\le j\le N_L} \|\rho(X^{(L-1)}W_j^{(L)}) - \rho(\widetilde{X}^{(L-1)}Q_j^{(L)})\|_2 \\
&\le \max_{1\le j\le N_L} \|X^{(L-1)}W_j^{(L)} - \widetilde{X}^{(L-1)}Q_j^{(L)}\|_2 \\
&\le \delta\sqrt{2\pi pm\log N_{L-1}} \max_{1\le j\le N_{L-1}} \|X_j^{(L-1)}\|_2 + \Big(N_{L-1}\|W^{(L)}\|_{\max}\|P^{(L-1)}\|_2^{r-1} \\
&+ \delta\sqrt{2\pi pm\log N_{L-1}}\Big) \max_{1\le j\le N_{L-1}} \|X^{(L-2)}W_j^{(L-1)} - \widetilde{X}^{(L-2)}Q_j^{(L-1)}\|_2
\end{aligned}$$

holds with probability exceeding $1 - \frac{\sqrt{2}mN_L}{N_{L-1}^p}$. Then (5.40) follows by inductively using part (b) of Theorem 5.3.3 with $i = L-1, L-2, \ldots, 1$. □

Remarks on the error bounds.

A few comments are in order regarding the error bounds associated with Corollary 5.3.4. First, let us consider the difference between the error bounds (5.39) and (5.40). As (5.40) deals with imperfect data alignment, it involves a term that bounds the mismatch between the quantized and unquantized networks. This term is controlled by the quantity $\|P^{(k)}\|_2^{r-1}$, which is expected to be small when the order r is sufficiently large provided $\|P^{(k)}\|_2 < 1$. In other words, one expects this term to be dominated by the error due to quantization. To get a sense for whether this intuition is valid, consider the case where $\widetilde{X}_1^{(k)}, \widetilde{X}_2^{(k)}, \ldots, \widetilde{X}_{N_k}^{(k)}$ are i.i.d. standard Gaussian vectors. Then Lemma 5.7.3 implies that, with high probability,

$$\|P^{(k)}\|_2^{r-1} \lesssim \left(1 - \frac{c}{m}\right)^{\frac{(r-1)N_k}{10}} = \left(1 - \frac{c}{m}\right)^{\frac{-m}{c}\cdot\frac{-c(r-1)N_k}{10m}} \le e^{-\frac{c(r-1)N_k}{10m}}$$

where $c > 0$ is a constant. In this case, $\|P^{(k)}\|_2^{r-1}$ decays exponentially with respect to r with a favorable dependence on the overparametrization $\frac{N}{m}$. In other words, here, even with a small order r, the error bounds in (5.39) and (5.40) are quite similar.

Keeping this in mind, our next objective is to assess the quality of these error bounds. We will accomplish this by examining the *relative error* connected to the quantization of a neural

network. Specifically, we will concentrate on evaluating the relative error associated with (5.39) since a similar derivation can be applied to (5.40).

We begin with the observation that both absolute error bounds (5.39) and (5.40) in Corollary 5.3.4 only involve randomness due to the stochastic quantizer $\mathscr{Q}_{\text{StocQ}}$. In particular, there is no randomness assumption on either the weights or the activations. However, to evaluate the relative error, we suppose that each $W^{(i)} \in \mathbb{R}^{N_{i-1}\times N_i}$ has i.i.d. $\mathscr{N}(0,1)$ entries and $\{W^{(i)}\}_{i=1}^{L}$ are independent. One needs to make an assumption of this type in order to facilitate the calculation, and more importantly, to avoid adversarial scenarios where the weights are chosen to be in the null-space of the data matrix $\widetilde{X}^{(i)}$. We obtain the following corollary which shows that the relative error decays with the overparametrization of the neural network.

Corollary 5.3.5. *Let Φ be an L-layer neural network as in* (5.3) *where the activation function is $\varphi^{(i)}(x) = \rho(x) := \max\{0,x\}$ for $1 \le i \le L$. Suppose the weight matrix $W^{(i)}$ has i.i.d. $\mathscr{N}(0,1)$ entries and $\{W^{(i)}\}_{i=1}^{L}$ are independent. Let $X \in \mathbb{R}^{m\times N_0}$ be the input data and $X^{(i)} = \Phi^{(i)}(X) \in \mathbb{R}^{m\times N_i}$ be the output of the i-th layer defined in* (5.11)*. Then the following inequalities hold.*
(a) Let $p \in \mathbb{N}$ with $p \ge 2$. For $1 \le i \le L$,

$$\max_{1\le j\le N_i} \|X_j^{(i)}\|_2 \le (4p)^{\frac{i}{2}} \Big(\prod_{k=1}^{i-1} N_k\Big)^{\frac{1}{2}} \Big(\prod_{k=0}^{i-1} \log N_k\Big)^{\frac{1}{2}} \|X\|_F \tag{5.41}$$

holds with probability at least $1 - \sum_{k=1}^{i} \frac{2N_k}{N_{k-1}^p}$.
(b) For $1 \le i \le L$, we have

$$\mathbb{E}_\Phi \|X^{(i)}\|_F^2 \ge \frac{\|X\|_F^2}{(2\pi)^i} \prod_{k=1}^{i} N_k \tag{5.42}$$

where $\mathbb{E}_\Phi$ denotes the expectation with respect to the weights of Φ, that is $\{W^{(i)}\}_{i=1}^{L}$.

Proof. (a) Conditioning on $X^{(i-1)}$, the function $f(z) := \|\rho(X^{(i-1)}z)\|_2$ is Lipschitz with Lipschitz constant $L_f := \|X^{(i-1)}\|_2 \le \|X^{(i-1)}\|_F$ and $\|X_j^{(i)}\|_2 = \|\rho(X^{(i-1)}W_j^{(i)})\|_2 = f(W_j^{(i)})$ with $W_j^{(i)} \sim \mathscr{N}(0,I)$. Applying Lemma 5.7.4 to f with $X = W_j^{(i)}$, Lipschitz constant L_f, and

$\alpha = \sqrt{2p\log N_{i-1}}\|X^{(i-1)}\|_F$, we obtain

$$\mathrm{P}\Big(\big|\|X_j^{(i)}\|_2 - \mathbb{E}(\|X_j^{(i)}\|_2 \mid X^{(i-1)})\big| \le \sqrt{2p\log N_{i-1}}\|X^{(i-1)}\|_F \,\Big|\, X^{(i-1)}\Big) \ge 1 - \frac{2}{N_{i-1}^p}. \quad (5.43)$$

Using Jensen's inequality and the identity $\mathbb{E}(\|\rho(X^{(i-1)}W_j^{(i)})\|_2^2 \mid X^{(i-1)}) = \frac{1}{2}\|X^{(i-1)}\|_F^2$, we have

$$\begin{aligned}\mathbb{E}(\|X_j^{(i)}\|_2 \mid X^{(i-1)}) &\le \Big(\mathbb{E}(\|X_j^{(i)}\|_2^2 \mid X^{(i-1)})\Big)^{\frac{1}{2}} \\ &= \Big(\mathbb{E}(\|\rho(X^{(i-1)}W_j^{(i)})\|_2^2 \mid X^{(i-1)})\Big)^{\frac{1}{2}} \\ &= \frac{1}{\sqrt{2}}\|X^{(i-1)}\|_F.\end{aligned}$$

It follows from the inequality above and (5.43) that, conditioning on $X^{(i-1)}$,

$$\|X_j^{(i)}\|_2 \le \Big(\frac{1}{\sqrt{2}} + \sqrt{2p\log N_{i-1}}\Big)\|X^{(i-1)}\|_F \le 2\sqrt{p\log N_{i-1}}\|X^{(i-1)}\|_F$$

holds with probability at least $1 - \frac{2}{N_{i-1}^p}$. Conditioning on $X^{(i-1)}$ and taking a union bound over $1 \le j \le N_i$, with probability exceeding $1 - \frac{2N_i}{N_{i-1}^p}$, we have

$$\|X^{(i)}\|_F \le \sqrt{N_i}\max_{1\le j\le N_i}\|X_j^{(i)}\|_2 \le 2\sqrt{pN_i\log N_{i-1}}\|X^{(i-1)}\|_F. \quad (5.44)$$

Applying (5.44) for indices $i, i-1, \ldots, 1$ recursively, we obtain (5.41).

(b) Applying Jensen's inequality and Proposition 5.7.5, we have

$$\begin{aligned}\mathbb{E}(\|X_j^{(i)}\|_2^2 \mid X^{(i-1)}) &= \mathbb{E}(\|\rho(X^{(i-1)}W_j^{(i)})\|_2^2 \mid X^{(i-1)}) \\ &\ge \Big(\mathbb{E}(\|\rho(X^{(i-1)}W_j^{(i)})\|_2 \mid X^{(i-1)})\Big)^2 \\ &\ge \frac{\mathrm{tr}(X^{(i-1)}X^{(i-1)\top})}{2\pi} \\ &= \frac{\|X^{(i-1)}\|_F^2}{2\pi}.\end{aligned}$$

By the law of total expectation, we obtain $\mathbb{E}_\Phi\|X_j^{(i)}\|_2^2 \geq \frac{1}{2\pi}\mathbb{E}_\Phi\|X^{(i-1)}\|_F^2$ and thus

$$\mathbb{E}_\Phi\|X^{(i)}\|_F^2 = \sum_{j=1}^{N_i} \mathbb{E}_\Phi\|X_j^{(i)}\|_2^2 \geq \frac{N_i}{2\pi}\mathbb{E}_\Phi\|X^{(i-1)}\|_F^2. \tag{5.45}$$

Then (5.42) follows immediately by applying (5.45) recursively. □

Now we are ready to evaluate the relative error associated with (5.39). It follows from (5.39) and the Cauchy-Schwarz inequality that, with high probability,

$$\begin{aligned}
\frac{\|\Phi(X)-\widetilde{\Phi}(X)\|_F^2}{\mathbb{E}_\Phi\|\Phi(X)\|_F^2} &\leq \frac{N_L \max_{1\leq j\leq N_L}\|\Phi(X)_j-\widetilde{\Phi}(X)_j\|_2^2}{\mathbb{E}_\Phi\|\Phi(X)\|_F^2} \\
&\leq \frac{N_L}{\mathbb{E}_\Phi\|\Phi(X)\|_F^2}\Big(\sum_{i=0}^{L-1}(2\pi pm\delta^2)^{\frac{L-i}{2}}\Big(\prod_{k=i}^{L-1}\log N_k\Big)^{\frac{1}{2}}\max_{1\leq j\leq N_i}\|X_j^{(i)}\|_2\Big)^2 \\
&\leq \frac{LN_L}{\mathbb{E}_\Phi\|\Phi(X)\|_F^2}\sum_{i=0}^{L-1}(2\pi pm\delta^2)^{L-i}\Big(\prod_{k=i}^{L-1}\log N_k\Big)\max_{1\leq j\leq N_i}\|X_j^{(i)}\|_2^2.
\end{aligned} \tag{5.46}$$

By Corollary 5.3.5, $\max_{1\leq j\leq N_i}\|X_j^{(i)}\|_2^2 \leq (4p)^i\|X\|_F^2\log N_0\prod_{k=1}^{i-1}(N_k\log N_k)$ with high probability, and $\mathbb{E}_\Phi\|\Phi(X)\|_F^2 = \mathbb{E}_\Phi\|X^{(L)}\|_F^2 \geq \frac{\|X\|_F^2}{(2\pi)^L}\prod_{k=1}^{L}N_k$. Plugging these results into (5.46),

$$\begin{aligned}
\frac{\|\Phi(X)-\widetilde{\Phi}(X)\|_F^2}{\mathbb{E}_\Phi\|\Phi(X)\|_F^2} &\leq L(2\pi)^L\Big(\prod_{k=0}^{L}\log N_k\Big)\sum_{i=0}^{L-1}\frac{(2\pi pm\delta^2)^{L-i}(4p)^i}{\prod_{k=i}^{L-1}N_k} \\
&\lesssim \Big(\prod_{k=0}^{L}\log N_k\Big)\sum_{i=0}^{L-1}\prod_{k=i}^{L-1}\frac{m}{N_k}
\end{aligned} \tag{5.47}$$

gives an upper bound on the relative error of quantization method in Algorithm 5. Further, if we assume $N_{\min}\leq N_i\leq N_{\max}$ for all i, and $2m\leq N_{\min}$, then (5.47) becomes

$$\begin{aligned}
\frac{\|\Phi(X)-\widetilde{\Phi}(X)\|_F^2}{\mathbb{E}_\Phi\|\Phi(X)\|_F^2} &\lesssim (\log N_{\max})^{L+1}\sum_{i=0}^{L-1}\Big(\frac{m}{N_{\min}}\Big)^{L-i} \\
&\lesssim \frac{m(\log N_{\max})^{L+1}}{N_{\min}}.
\end{aligned}$$

This high probability estimate indicates that the squared error resulting from quantization decays

with the overparametrization of the network, relative to the expected squared norm of the neural network's output. It may be possible to replace the expected squared norm by the squared norm itself using another high probability estimate. However, we refrain from doing so as the main objective of this computation was to gain insight into the decay of the relative error in generic settings and the expectation suffices for that purpose.

5.4 Error Bounds for SPFQ with Finite Alphabets

Our goal for this section is to relax the assumption that the quantization alphabet used in our algorithms is infinite. We would also like to evaluate the number of elements $2K$ in our alphabet, and thus the number of bits $b := \log_2(K) + 1$ needed for quantizing each layer. Moreover, for simplicity, here we will only consider Algorithm 5. In this setting, to use a finite quantization alphabet, and still obtain theoretical error bounds, we must guarantee that the argument of the stochastic quantizer in (5.19) remains smaller than the maximal element in the alphabet. Indeed, if that is the case for all $t = 1, ..., N_{i-1}$ then the error bound for our finite alphabet would be identical as for the infinite alphabet. It remains to determine the right size of such a finite alphabet. To that end, we start with Theorem 5.4.1, which assumes boundedness of all the aligned weights $\widetilde{w}$ in the i-th layer, i.e., the solutions of (5.22), in order to generate an error bound for a finite alphabet of size $K^{(i)} \gtrsim \sqrt{\log \max\{N_{i-1}, N_i\}}$.

Theorem 5.4.1. *Assuming that the first $i-1$ layers have been quantized, let $X^{(i-1)}$, $\widetilde{X}^{(i-1)}$ be as in* (5.11). *Let $p, K^{(i)} \in \mathbb{N}$ and $\delta > 0$ satisfying $p \geq 3$. Suppose we quantize $W^{(i)}$ using Algorithm 5 with $\mathscr{A} = \mathscr{A}^{\delta}_{K^{(i)}}$ and suppose the resulting aligned weights $\widetilde{W}^{(i)}$ from solving* (5.22) *satisfy*

$$\|\widetilde{W}^{(i)}\|_{\max} \leq \frac{1}{2} K^{(i)} \delta. \tag{5.48}$$

Then

$$\max_{1 \leq j \leq N_i} \|X^{(i-1)} W_j^{(i)} - \widetilde{X}^{(i-1)} Q_j^{(i)}\|_2 \leq \delta \sqrt{2\pi p m \log N_{i-1}} \max_{1 \leq j \leq N_{i-1}} \|\widetilde{X}_j^{(i-1)}\|_2 \tag{5.49}$$

holds with probability at least $1-\frac{\sqrt{2}mN_i}{N_{i-1}^p}-\sqrt{2}N_i\sum_{t=2}^{N_{i-1}}\exp\Big(-\frac{(K^{(i)})^2\|\widetilde{X}_t^{(i-1)}\|_2^2}{8\pi\max_{1\le j\le t-1}\|\widetilde{X}_j^{(i-1)}\|_2^2}\Big)$.

Proof. Fix a neuron $w := W_j^{(i)} \in \mathbb{R}^{N_{i-1}}$ for some $1 \le j \le N_i$. By our assumption (5.48), the aligned weights $\widetilde{w}$ satisfy $\|\widetilde{w}\|_\infty \le \frac{1}{2}K^{(i)}\delta$. Then, we perform the iteration (5.19) in Algorithm 5. At the t-th step, similar to (5.28), (5.30), and (5.32), we have

$$\widetilde{u}_t = P_{\widetilde{X}_t^{(i-1)\perp}}(h_t) + (v_t - \widetilde{q}_t)\widetilde{X}_t^{(i-1)}$$

where

$$h_t = \widetilde{u}_{t-1} + \widetilde{w}_t\widetilde{X}_t^{(i-1)}, \quad v_t = \frac{\langle \widetilde{X}_t^{(i-1)}, h_t\rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2}, \quad \text{and} \quad \widetilde{q}_t = \mathscr{Q}_{\mathrm{StocQ}}(v_t). \tag{5.50}$$

If $t=1$, then $h_1 = \widetilde{w}_1\widetilde{X}_1^{(i-1)}$, $v_1 = \widetilde{w}_1$, and $\widetilde{q}_1 = \mathscr{Q}_{\mathrm{StocQ}}(v_1)$. Since $|v_1| = |\widetilde{w}_1| \le \|\widetilde{w}\|_\infty \le \frac{1}{2}K^{(i)}\delta$, we get $|v_1 - \widetilde{q}_1| \le \delta$ and the proof technique used for the case $t=1$ in Lemma 5.3.1 can be applied here to conclude that $\widetilde{u}_1 \le_{\mathrm{cx}} \mathscr{N}(0,\sigma_1^2 I)$ with $\sigma_1^2 = \frac{\pi\delta^2}{2}\|\widetilde{X}_1^{(i-1)}\|_2^2$. Next, for $t \ge 2$, assume that $\widetilde{u}_{t-1} \le_{\mathrm{cx}} \mathscr{N}(0,\sigma_{t-1}^2 I)$ holds where $\sigma_{t-1}^2 = \frac{\pi\delta^2}{2}\max_{1\le j\le t-1}\|\widetilde{X}_j^{(i-1)}\|_2^2$ is defined as in Lemma 5.3.1. It follows from (5.50) and Lemma 5.6.3 that

$$|v_t| = \Big|\frac{\langle \widetilde{X}_t^{(i-1)}, \widetilde{u}_{t-1}\rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2} + \widetilde{w}_t\Big| \le \Big|\frac{\langle \widetilde{X}_t^{(i-1)}, \widetilde{u}_{t-1}\rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2}\Big| + \|\widetilde{w}\|_\infty \le \Big|\frac{\langle \widetilde{X}_t^{(i-1)}, \widetilde{u}_{t-1}\rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2}\Big| + \frac{1}{2}K^{(i)}\delta$$

with $\frac{\langle \widetilde{X}_t^{(i-1)}, \widetilde{u}_{t-1}\rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2} \le_{\mathrm{cx}} \mathscr{N}\Big(0, \frac{\sigma_{t-1}^2}{\|\widetilde{X}_t^{(i-1)}\|_2^2}\Big)$. Then we have, by Lemma 5.7.2, that

$$\mathrm{P}(|v_t| \le K^{(i)}\delta) \ge \mathrm{P}\Big(\Big|\frac{\langle \widetilde{X}_t^{(i-1)}, \widetilde{u}_{t-1}\rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2}\Big| \le \frac{1}{2}K^{(i)}\delta\Big) \ge 1 - \sqrt{2}\exp\Big(-\frac{(K^{(i)}\delta)^2}{16\sigma_{t-1}^2}\|\widetilde{X}_t^{(i-1)}\|_2^2\Big).$$

On the event $\{|v_t| \le K^{(i)}\delta\}$, we can quantize v_t as if the quantizer $\mathscr{Q}_{\mathrm{StocQ}}$ used the infinite alphabet $\mathscr{A}_\infty^\delta$. So $\widetilde{u}_t \le_{\mathrm{cx}} \mathscr{N}(0,\sigma_t^2 I)$. Therefore, applying a union bound,

$$\mathrm{P}\Big(\widetilde{u}_{N_{i-1}} \le_{\mathrm{cx}} \mathscr{N}(0,\sigma_{N_{i-1}}^2 I)\Big) \ge 1 - \sqrt{2}\sum_{t=2}^{N_{i-1}}\exp\Big(-\frac{(K^{(i)}\delta)^2}{16\sigma_{t-1}^2}\|\widetilde{X}_t^{(i-1)}\|_2^2\Big). \tag{5.51}$$

Conditioning on the event above, that $\widetilde{u}_{N_{i-1}} \leq_{\text{cx}} \mathcal{N}(0, \sigma^2_{N_{i-1}} I)$, Lemma 5.7.2 yields for $\gamma \in (0,1]$

$$\mathrm{P}\Big(\|\widetilde{u}_{N_{i-1}}\|_\infty \leq 2\sigma_{N_{i-1}} \sqrt{\log(\sqrt{2}m/\gamma)} \Big) \geq 1-\gamma.$$

Setting $\gamma = \sqrt{2} m N_{i-1}^{-p}$ and recalling (5.23), we obtain that

$$\|X^{(i-1)} W_j^{(i)} - \widetilde{X}^{(i-1)} Q_j^{(i)}\|_2 = \|\widetilde{u}_{N_{i-1}}\|_2 \leq \sqrt{m} \|\widetilde{u}_{N_{i-1}}\|_\infty \leq 2\sigma_{N_{i-1}} \sqrt{mp \log N_{i-1}} \tag{5.52}$$

holds with probability at least $1 - \frac{\sqrt{2}m}{N_{i-1}^p}$. Combining (5.51) and (5.52), for each $1 \leq j \leq N_i$,

$$\|X^{(i-1)} W_j^{(i)} - \widetilde{X}^{(i-1)} Q_j^{(i)}\|_2 \leq 2\sigma_{N_{i-1}} \sqrt{mp \log N_{i-1}} = \delta \sqrt{2\pi pm \log N_{i-1}} \max_{1 \leq j \leq N_{i-1}} \|\widetilde{X}_j^{(i-1)}\|_2$$

holds with probability exceeding $1 - \frac{\sqrt{2}m}{N_{i-1}^p} - \sqrt{2} \sum_{t=2}^{N_{i-1}} \exp\Big(-\frac{(K^{(i)}\delta)^2}{16\sigma_{t-1}^2} \|\widetilde{X}_t^{(i-1)}\|_2^2 \Big)$. Taking a union bound over all $1 \leq j \leq N_i$, we have

$$\begin{aligned}
&\mathrm{P}\Big(\max_{1 \leq j \leq N_i} \|X^{(i-1)} W_j^{(i)} - \widetilde{X}^{(i-1)} Q_j^{(i)}\|_2 \leq \delta \sqrt{2\pi pm \log N_{i-1}} \max_{1 \leq j \leq N_{i-1}} \|\widetilde{X}_j^{(i-1)}\|_2 \Big) \\
&\geq 1 - \frac{\sqrt{2}mN_i}{N_{i-1}^p} - \sqrt{2} N_i \sum_{t=2}^{N_{i-1}} \exp\Big(-\frac{(K^{(i)}\delta)^2}{16\sigma_{t-1}^2} \|\widetilde{X}_t^{(i-1)}\|_2^2 \Big) \\
&\geq 1 - \frac{\sqrt{2}mN_i}{N_{i-1}^p} - \sqrt{2} N_i \sum_{t=2}^{N_{i-1}} \exp\Big(-\frac{(K^{(i)})^2 \|\widetilde{X}_t^{(i-1)}\|_2^2}{8\pi \max_{1 \leq j \leq t-1} \|\widetilde{X}_j^{(i-1)}\|_2^2} \Big).
\end{aligned}$$

□

Next, in Theorem 5.4.2, we show that provided the activations $X^{(i-1)}$ and $\widetilde{X}^{(i-1)}$ of the quantized and unquantized networks are sufficiently close, and provided the weights w follow a random distribution, one can guarantee the needed boundedness of the aligned weights $\widetilde{w}$. This allows us to apply Theorem 5.4.1 and generate an error bound for finite alphabets. Our focus on random weights here enables us to avoid certain adversarial situations. Indeed, one can construct activations $X^{(i-1)}$ and $\widetilde{X}^{(i-1)}$ that are arbitrarily close to each other, along with

adversarial weights w that together lead to $\|\widetilde{w}\|_\infty$ becoming arbitrarily large. We demonstrate this contrived adversarial scenario in Proposition 5.8.1. However, in generic cases represented by random weights, as shown in Theorem 5.4.2, the bound on $\widetilde{w}$ is not a major issue. Consequently, one can utilize a finite alphabet for quantization as desired.

Theorem 5.4.2. *Assuming that the first $i-1$ layers have been quantized, let $X^{(i-1)}$, $\widetilde{X}^{(i-1)}$ be as in* (5.11)*. Suppose the weight matrix $W^{(i)} \in \mathbb{R}^{N_{i-1}\times N_i}$ has i.i.d. $\mathscr{N}(0,1)$ entries and*

$$\|\widetilde{X}^{(i-1)} - X^{(i-1)}\|_2 \leq \varepsilon^{(i-1)}\sigma_1^{(i-1)} < \sigma_m^{(i-1)}, \tag{5.53}$$

where $\varepsilon^{(i-1)} \in (0,1)$, $\sigma_1^{(i-1)}$ and $\sigma_m^{(i-1)}$ are the largest and smallest singular values of $X^{(i-1)}$ respectively. Let $p, K^{(i)} \in \mathbb{N}$ and $\delta > 0$ such that $p \geq 3$ and

$$K^{(i)}\delta \geq 2\eta^{(i-1)}\sqrt{2p\log N_{i-1}}. \tag{5.54}$$

where $\eta^{(i-1)} := \frac{\sigma_1^{(i-1)}}{\sigma_m^{(i-1)} - \varepsilon^{(i-1)}\sigma_1^{(i-1)}}$. If we quantize $W^{(i)}$ using Algorithm 5 with $\mathscr{A} = \mathscr{A}^{\delta}_{K^{(i)}}$, then

$$\max_{1\leq j\leq N_i} \|X^{(i-1)}W_j^{(i)} - \widetilde{X}^{(i-1)}Q_j^{(i)}\|_2 \leq \delta\sqrt{2\pi pm\log N_{i-1}} \max_{1\leq j\leq N_{i-1}} \|\widetilde{X}_j^{(i-1)}\|_2 \tag{5.55}$$

holds with probability at least $1 - \frac{2N_i}{N_{i-1}^{p-1}} - \frac{\sqrt{2}mN_i}{N_{i-1}^{p}} - \sqrt{2}N_i\sum_{t=2}^{N_{i-1}} \exp\Big(-\frac{(K^{(i)})^2\|\widetilde{X}_t^{(i-1)}\|_2^2}{8\pi\max_{1\leq j\leq t-1}\|\widetilde{X}_j^{(i-1)}\|_2^2}\Big)$.

Proof. Pick a neuron $w := W_j^{(i)} \in \mathbb{R}^{N_{i-1}}$ for some $1 \leq j \leq N_i$. Then we have $w \sim \mathscr{N}(0, I)$ and since we are using Algorithm 5, we must work with the resulting $\widetilde{w}$, the solution of (5.22). Applying Proposition 5.8.3 to w with $X = X^{(i-1)}$ and $\widetilde{X} = \widetilde{X}^{(i-1)}$, we obtain

$$\mathrm{P}\Big(\|\widetilde{w}\|_\infty \leq \eta^{(i-1)}\sqrt{2p\log N_{i-1}}\Big) \geq 1 - \frac{2}{N_{i-1}^{p-1}},$$

so that using (5.54) gives

$$\mathrm{P}\Big(\|\widetilde{w}\|_\infty \leq \frac{1}{2}K^{(i)}\delta\Big) \geq 1 - \frac{2}{N_{i-1}^{p-1}}. \tag{5.56}$$

Conditioning on the event $\{\|\widetilde{w}\|_\infty \leq \frac{1}{2}K^{(i)}\delta\}$ and applying exactly the same argument in Theorem 5.4.1,

$$\|X^{(i-1)}W_j^{(i)} - \widetilde{X}^{(i-1)}Q_j^{(i)}\|_2 \leq \delta\sqrt{2\pi pm\log N_{i-1}}\max_{1\leq j\leq N_{i-1}}\|\widetilde{X}_j^{(i-1)}\|_2 \tag{5.57}$$

holds with probability exceeding $1-\frac{\sqrt{2}m}{N_{i-1}^p}-\sqrt{2}\sum_{t=2}^{N_{i-1}}\exp\Big(-\frac{(K^{(i)})^2\|\widetilde{X}_t^{(i-1)}\|_2^2}{8\pi\max_{1\leq j\leq t-1}\|\widetilde{X}_j^{(i-1)}\|_2^2}\Big)$. Combining (5.56) and (5.57), and taking a union bound over all $1\leq j\leq N_i$, we obtain (5.55). □

Now we are about to approximate the number of bits needed for guaranteeing the derived bounds. Note that, in Theorem 5.4.2, we achieved the same error bound (5.55) as in Lemma 5.3.1, choosing proper $\varepsilon^{(i-1)}\in(0,1)$ and $K^{(i)}\in\mathbb{N}$ such that (5.53) and (5.54) are satisfied and the associated probability in (5.55) is positive. This implies that the error bounds we obtained in Section 5.3 remain valid for our finite alphabets as well. In particular, by a similar argument we used to obtain (5.47), one can get the following approximations

$$\frac{\|\widetilde{X}^{(i-1)}-X^{(i-1)}\|_F^2}{\|X^{(i-1)}\|_F^2}\lesssim\Big(\prod_{k=0}^{i-1}\log N_k\Big)\sum_{j=0}^{i-2}\prod_{k=j}^{i-2}\frac{m}{N_k}.$$

Due to $\|X^{(i-1)}\|_F\leq\sqrt{m}\|X^{(i-1)}\|_2$ and $\|\widetilde{X}^{(i-1)}-X^{(i-1)}\|_2\leq\|\widetilde{X}^{(i-1)}-X^{(i-1)}\|_F$, we have

$$\begin{aligned}\frac{\|\widetilde{X}^{(i-1)}-X^{(i-1)}\|_2^2}{\|X^{(i-1)}\|_2^2}&\leq\frac{m\|\widetilde{X}^{(i-1)}-X^{(i-1)}\|_F^2}{\|X^{(i-1)}\|_F^2}\\&\lesssim m\Big(\prod_{k=0}^{i-1}\log N_k\Big)\sum_{j=0}^{i-2}\prod_{k=j}^{i-2}\frac{m}{N_k}.\end{aligned}$$

If $\prod_{k=j}^{i-2}N_k\gtrsim m^{i-j}\prod_{k=0}^{i-1}\log N_k$ for $0\leq j\leq i-2$, then it is possible to choose $\varepsilon^{(i-1)}\in(0,1)$ such that (5.53) holds. Moreover, since $\sigma_m^{(i-1)}\leq\sigma_1^{(i-1)}$, we have $\eta^{(i-1)}=\frac{\sigma_1^{(i-1)}}{\sigma_m^{(i-1)}-\varepsilon^{(i-1)}\sigma_1^{(i-1)}}\geq(1-\varepsilon^{(i-1)})^{-1}$ and thus (5.54) becomes

$$K^{(i)}\geq 2\delta^{-1}(1-\varepsilon^{(i-1)})^{-1}\sqrt{2p\log N_{i-1}}\gtrsim\sqrt{\log N_{i-1}}. \tag{5.58}$$

Assuming columns of $\widetilde{X}^{(i-1)}$ are similar in the sense of

$$\max_{1\le j\le t-1} \|\widetilde{X}_j^{(i-1)}\|_2 \lesssim \sqrt{\log N_{i-1}}\|\widetilde{X}_t^{(i-1)}\|_2, \quad 2\le t\le N_{i-1},$$

we obtain that (5.55) holds with probability exceeding

$$\begin{aligned} &1-\frac{2N_i}{N_{i-1}^{p-1}}-\frac{\sqrt{2}mN_i}{N_{i-1}^{p}}-\sqrt{2}N_i\sum_{t=2}^{N_{i-1}}\exp\Big(-\frac{(K^{(i)})^2\|\widetilde{X}_t^{(i-1)}\|_2^2}{8\pi\max_{1\le j\le t-1}\|\widetilde{X}_j^{(i-1)}\|_2^2}\Big)\\ &\ge 1-\frac{2N_i}{N_{i-1}^{p-1}}-\frac{\sqrt{2}mN_i}{N_{i-1}^{p}}-\sqrt{2}N_{i-1}N_i\exp\Big(-\frac{(K^{(i)})^2}{8\pi\log N_{i-1}}\Big). \end{aligned} \tag{5.59}$$

To make (5.59) positive, we have

$$K^{(i)} \gtrsim \log\max\{N_{i-1},N_i\}. \tag{5.60}$$

It follows from (5.58) and (5.59) that, in the ith layer, we only need a number of bits $b^{(i)}$ that satisfies

$$b^{(i)} \ge \log_2 K^{(i)}+1 \gtrsim \log_2\log\max\{N_{i-1},N_i\}$$

to guarantee the performance of our quantization method using finite alphabets.

Table 5.1. Top-1/Top-5 validation accuracy for SPFQ on ImageNet.

Model	m	b	C	Quant Acc (%)	Ref Acc (%)	Acc Drop (%)
VGG-16	1024	4	1.02	70.48/89.77	71.59/90.38	1.11/0.61
		5	1.23	71.08/90.15	71.59/90.38	0.51/0.23
		6	1.26	71.24/90.37	71.59/90.38	0.35/0.01
ResNet-18	2048	4	0.91	67.36/87.74	69.76/89.08	2.40/1.34
		5	1.32	68.79/88.77	69.76/89.08	0.97/0.31
		6	1.68	69.43/88.96	69.76/89.08	0.33/0.12
ResNet-50	2048	4	1.10	73.37/91.61	76.13/92.86	2.76/1.25
		5	1.62	75.05/92.43	76.13/92.86	1.08/0.43
		6	1.98	75.66/92.67	76.13/92.86	0.47/0.19

5.5 Experiments

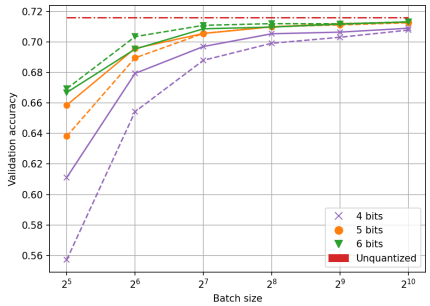
In this section, we test the performance of SPFQ on the ImageNet classification task and compare it with the non-random scheme GPFQ in [38]. In particular, we adopt the version of SPFQ corresponding to (5.15) [1], i.e., Algorithm 6 with order $r = 1$. Note that the GPFQ algorithm runs the same iterations as in (5.15) except that $\mathscr{Q}_{\text{StocQ}}$ is substituted with a non-random quantizer $\mathscr{Q}_{\text{DetQ}}$, so the associated iterations are given by

$$\begin{cases} u_0 = 0 \in \mathbb{R}^m, \\ q_t = \mathscr{Q}_{\text{DetQ}}\left(\frac{\langle \widetilde{X}_t^{(i-1)}, u_{t-1} + w_t X_t^{(i-1)} \rangle}{\|\widetilde{X}_t^{(i-1)}\|_2^2}\right), \\ u_t = u_{t-1} + w_t X_t^{(i-1)} - q_t \widetilde{X}_t^{(i-1)} \end{cases} \tag{5.61}$$

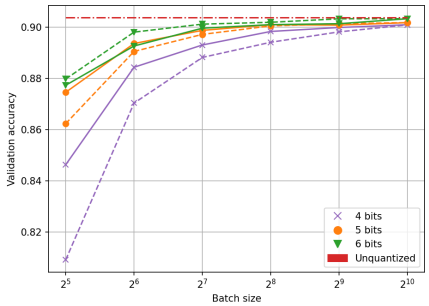
where $\mathscr{Q}_{\text{DetQ}}(z) := \text{argmin}_{p \in \mathscr{A}} |z - p|$. For ImageNet data, we consider ILSVRC-2012 [11], a 1000-category dataset with over 1.2 million training images and 50 thousand validation images. Additionally, we resize all images to 256×256 and use the normalized 224×224 center crop, which is a standard procedure. The evaluation metrics we choose are top-1 and top-5 accuracy of the quantized models on the validation dataset. As for the neural network architectures, we quantize all layers of VGG-16 [33], ResNet-18 and ResNet-50 [19], which are pretrained 32-bit floating point neural networks provided by torchvision in PyTorch [28]. Moreover, we fuse the batch normalization (BN) layer with the convolutional layer, and freeze the BN statistics before quantization.

Since the major difference between SPFQ in (5.15) and GPFQ in (5.61) is the choice of quantizers, we will follow the experimental setting for alphabets used in [38]. Specifically, we use batch size m, fixed bits $b \in \mathbb{N}$ for all the layers, and quantize each $W^{(i)} \in \mathbb{R}^{N_{i-1} \times N_i}$ with midtread alphabets $\mathscr{A} = \mathscr{A}_K^\delta$ as in (5.5), where level K and step size δ are given by

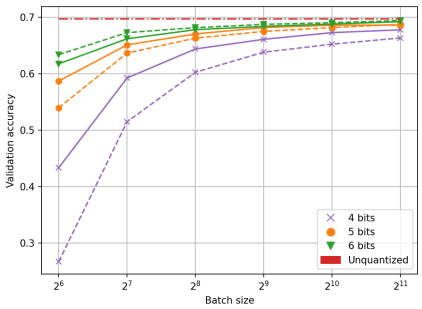
[1] Code: https : //github.com/jayzhang0727/Stochastic – Path – Following – Quantization.git



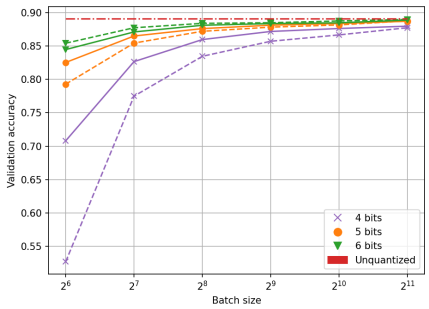
(a) Top-1 accuracy of VGG-16



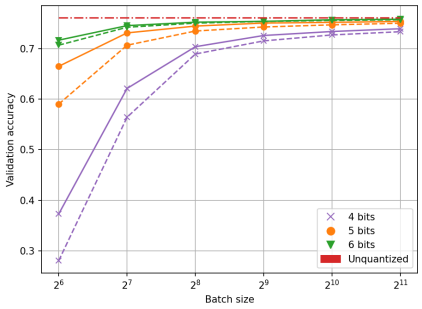
(b) Top-5 accuracy of VGG-16



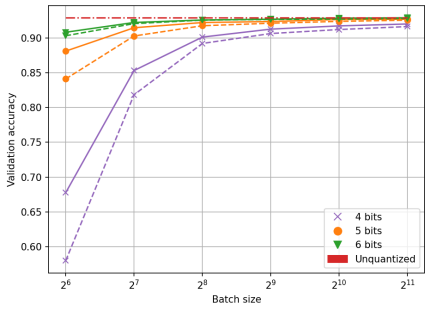
(c) Top-1 accuracy of ResNet-18



(d) Top-5 accuracy of ResNet-18



(e) Top-1 accuracy of ResNet-50



(f) Top-5 accuracy of ResNet-50

Figure 5.1. Top-1 and Top-5 validation accuracy for SPFQ (dashed lines) and GPFQ (solid lines) on ImageNet.

$$K = 2^{b-1}, \quad \delta = \delta^{(i)} := \frac{C}{2^{b-1}N_i} \sum_{1 \le j \le N_i} \|W_j^{(i)}\|_\infty.$$

Here, $C > 0$ is a constant that is only dependent on bitwidth b, determined by grid search with cross-validation, and fixed across layers, and across batch-sizes. One can, of course, expect to do better by using different values of C for different layers but we refrain from doing so, as our

main goal here is to demonstrate the performance of SPFQ even with minimal fine-tuning.

In Table 5.1, for different combinations of m, b, and C, we present the corresponding top-1/top-5 validation accuracy of quantized networks using SPFQ in the first column, while the second and thrid columns give the validation accuracy of unquantized models and the accuracy drop due to quantization respectively. We observe that, for all three models, the quantization accuracy is improved as the number of bits b increases, and SPFQ achieves less than 0.5% top-1 accuracy loss while using 6 bits.

Next, in Figure 5.1, we compare SPFQ against GPFQ by quantizing the three models in Table 5.1. These figures illustrate that GPFQ has better performance than that of SPFQ when $b = 3,4$ and m is small. This is not particularly surprising, as $\mathscr{Q}_{\text{DetQ}}$ deterministically rounds its argument to the nearest alphabet element instead of performing a random rounding like $\mathscr{Q}_{\text{StocQ}}$. However, as the batch size m increases, the accuracy gap between GPFQ and SPFQ diminishes. Indeed, for VGG-16 and ResNet-18, SPFQ outperforms GPFQ when $b = 6$. Further, we note that, for both SPFQ and GPFQ, one can obtain higher quantization accuracy by taking larger m but the extra improvement that results from increasing the batch size rapidly decreases.

5.6 Properties of Convex Orders

Throughout this section, $\overset{d}{=}$ denotes equality in distribution. A well-known result is that the convex order can be characterized by a *coupling* of X and Y, i.e. constructing X and Y on the same probability space.

Theorem 5.6.1 (Theorem 7.A.1 in [32])**.** *The random vectors X and Y satisfy $X \leq_{\text{cx}} Y$ if and only if there exist two random vectors $\hat{X}$ and $\hat{Y}$, defined on the same probability space, such that $\hat{X} \overset{d}{=} X$, $\hat{Y} \overset{d}{=} Y$, and $\mathbb{E}(\hat{Y}|\hat{X}) = \hat{X}$.*

In Theorem 5.6.1, $\mathbb{E}(\hat{Y}|\hat{X}) = \hat{X}$ implies $\mathbb{E}(\hat{Y} - \hat{X}|\hat{X}) = 0$. Let $\hat{Z} := \hat{Y} - \hat{X}$. Then we have $\hat{Y} = \hat{X} + \hat{Z}$ with $\mathbb{E}(\hat{Z}|\hat{X}) = 0$. Thus, one can obtain $\hat{Y}$ by first sampling $\hat{X}$, and then adding a mean 0 random vector $\hat{Z}$ whose distribution may depend on the sampled $\hat{X}$. Based on this important

observation, the following result gives necessary and sufficient conditions for the comparison of multivariate normal random vectors, see e.g. Example 7.A.13 in [32].

Lemma 5.6.2. *Consider multivariate normal distributions $\mathcal{N}(\mu_1,\Sigma_1)$ and $\mathcal{N}(\mu_2,\Sigma_2)$. Then*

$$\mathcal{N}(\mu_1,\Sigma_1) \leq_{\text{cx}} \mathcal{N}(\mu_2,\Sigma_2) \iff \mu_1 = \mu_2 \quad \textit{and} \quad \Sigma_1 \preceq \Sigma_2.$$

Proof. ($\Rightarrow$) Suppose that $X \sim \mathcal{N}(\mu_1,\Sigma_1)$ and $Y \sim \mathcal{N}(\mu_2,\Sigma_2)$ such that $X \leq_{\text{cx}} Y$. By (5.10), we have $\mu_1 = \mu_2$. Let $a \in \mathbb{R}^n$ and define $f(x) := (a^\top x - a^\top \mu_1)^2$. Since $f(x)$ is convex, one can get

$$a^\top \Sigma_1 a = \text{Var}(a^\top X) = \mathbb{E} f(X) \leq \mathbb{E} f(Y) = \text{Var}(a^\top Y) = a^\top \Sigma_2 a.$$

Since this inequality holds for arbitrary $a \in \mathbb{R}^n$, we obtain $\Sigma_1 \preceq \Sigma_2$.
($\Leftarrow$) Conversely, assume that $\mu_1 = \mu_2$ and $\Sigma_1 \preceq \Sigma_2$. Let $X \sim \mathcal{N}(\mu_1,\Sigma_1)$ and $Z \sim \mathcal{N}(0,\Sigma_2 - \Sigma_1)$ be independent. Construct a random vector $Y := X + Z$. Then $Y \sim \mathcal{N}(\mu_2,\Sigma_2)$ and $\mathbb{E}(Y|X) = \mathbb{E}(X+Z|X) = X + \mathbb{E}Z = X$. Following Theorem 5.6.1, $\mathcal{N}(\mu_1,\Sigma_1) \leq_{\text{cx}} \mathcal{N}(\mu_2,\Sigma_2)$ holds. □

Moreover, the convex order is preserved under affine transformations.

Lemma 5.6.3. *Suppose that X, Y are n-dimensional random vectors satisfying $X \leq_{\text{cx}} Y$. Let $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. Then $AX + b \leq_{\text{cx}} AY + b$.*

Proof. Let $f : \mathbb{R}^n \to \mathbb{R}$ be any convex function. Since $g(x) := f(Ax+b)$ is a composition of convex function $f(x)$ and a linear map, $g(x)$ is also convex. As $X \leq_{\text{cx}} Y$, we now have

$$\mathbb{E} f(AX+b) = \mathbb{E} g(X) \leq \mathbb{E} g(Y) = \mathbb{E} f(AY+b),$$

so $AX + b \leq_{\text{cx}} AY + b$. □

The following results, which will also be useful to us, were proved in Section 2 of [3].

Lemma 5.6.4. *Consider random vectors X, Y, W, and Z. Let X and Y live on the same probability space, and let W and Z be independent. Suppose that $X \leq_{\text{cx}} W$ and $(Y-X)|X \leq_{\text{cx}} Z$. Then $Y \leq_{\text{cx}} W+Z$.*

Lemma 5.6.5. *Let X be a real-valued random variable with $\mathbb{E}X = 0$ and $|X| \leq C$. Then $X \leq_{\text{cx}} \mathcal{N}\left(0, \frac{\pi C^2}{2}\right)$.*

Applying Lemma 5.6.4 inductively, one can show that the convex order is closed under convolutions.

Lemma 5.6.6. *Let $X_1, X_2, \ldots, X_m$ be a set of independent random vectors and let $Y_1, Y_2, \ldots, Y_m$ be another set of independent random vectors. If $X_i \leq_{\text{cx}} Y_i$ for $1 \leq i \leq m$, then*

$$\sum_{i=1}^{m} X_i \leq_{\text{cx}} \sum_{i=1}^{m} Y_i. \tag{5.62}$$

Proof. We will prove (5.62) by induction on m. The case $m = 1$ is trivial. Assume that the lemma holds for $m-1$ with $m \geq 2$, and let us prove it for m. Applying Lemma 5.6.4 for $X = X_m$, $Y = \sum_{i=1}^{m} X_i$, $W = Y_m$, and $Z = \sum_{i=1}^{m-1} Y_i$, inequality (5.62) follows. □

5.7 Useful Lemmata

The following two lemmata are essential for the approximation of quantization error bounds. The proof techniques follow [3].

Lemma 5.7.1. *Let $\alpha > 0$ and $z_1, z_2, \ldots, z_d \in \mathbb{R}^m$ be nonzero vectors. Let $M_0 = 0$. For $1 \leq t \leq d$, define $M_t \in \mathbb{R}^{m \times m}$ inductively as*

$$M_t := P_{z_t^\perp} M_{t-1} P_{z_t^\perp} + \alpha z_t z_t^\top$$

where $P_{z_t^\perp} = I - \frac{z_t z_t^\top}{\|z_t\|_2^2}$ is the orthogonal projection as in (5.8)*. Then*

$$M_t \preceq \beta_t I \tag{5.63}$$

holds for all t, where $\beta_t := \alpha \max_{1\le j\le t} \|z_j\|_2^2$.

Proof. We proceed by induction on t. If $t = 1$, then $M_1 = \alpha z_1 z_1^\top$. By Cauchy-Schwarz inequality, for any $x \in \mathbb{R}^m$, we get

$$x^\top M_1 x = \alpha \langle z_1, x\rangle^2 \le \alpha \|z_1\|_2^2 \|x\|_2^2 = \beta_1 \|x\|_2^2 = x^\top(\beta_1 I)x.$$

It follows that $M_1 \preceq \beta_1 I$. Now, assume that (5.63) holds for $t-1$ with $t \ge 2$. Then we have

$$\begin{aligned}
M_t &= P_{z_t^\perp} M_{t-1} P_{z_t^\perp} + \alpha z_t z_t^\top \\
&\preceq \beta_{t-1} P_{z_t^\perp}^2 + \alpha z_t z_t^\top && (\text{by assumption } M_{t-1} \preceq \beta_{t-1} I) \\
&\preceq \beta_t P_{z_t^\perp} + \alpha z_t z_t^\top && (\text{since } P_{z_t^\perp}^2 = P_{z_t^\perp} \text{ and } \beta_{t-1} \le \beta_t) \\
&= \beta_t I + (\alpha \|z_t\|_2^2 - \beta_t)\frac{z_t z_t^\top}{\|z_t\|_2^2} && (\text{using (5.8)}) \\
&\preceq \beta_t I && (\text{as } \beta_t = \alpha \max_{1\le j\le t} \|z_j\|_2^2).
\end{aligned}$$

This completes the proof. □

Lemma 5.7.2. *Let X be an n-dimensional random vector such that $X \le_{\text{cx}} \mathcal{N}(\mu, \sigma^2 I)$, and let $\alpha > 0$. Then*

$$\mathrm{P}\Big(\|X - \mu\|_\infty \le \alpha \Big) \ge 1 - \sqrt{2} n e^{-\frac{\alpha^2}{4\sigma^2}}.$$

In particular, if $\alpha = 2\sigma\sqrt{\log(\sqrt{2}n/\gamma)}$ with $\gamma \in (0,1]$, we have

$$P\left(\|X - \mu\|_\infty \le 2\sigma\sqrt{\log(\sqrt{2}n/\gamma)} \right) \ge 1 - \gamma.$$

Proof. Let $x \in \mathbb{R}^n$ with $\|x\|_2 \le 1$. Since $X \le_{\text{cx}} \mathcal{N}(\mu, \sigma^2 I)$, by Lemma 5.6.2 and Lemma 5.6.3, we get

$$\frac{\langle X - \mu, x\rangle}{\sigma} \le_{\text{cx}} \mathcal{N}(0, \|x\|_2^2) \le_{\text{cx}} \mathcal{N}(0,1).$$

Then we have

$$\mathbb{E}e^{\frac{\langle X-\mu,x\rangle^2}{4\sigma^2}} \leq \mathbb{E}_{Z\sim\mathcal{N}(0,1)}e^{Z^2/4} = \sqrt{2}.$$

where we used Definition 5.2.1 on the convex function $f(x) = e^{x^2/4}$. By Markov's inequality and the inequality above, we conclude that

$$\begin{aligned}
\mathrm{P}(|\langle X-\mu,x\rangle| \geq \alpha) &= \mathrm{P}\Big(e^{\frac{\langle X-\mu,x\rangle^2}{4\sigma^2}} \geq e^{\frac{\alpha^2}{4\sigma^2}}\Big) \\
&\leq e^{-\frac{\alpha^2}{4\sigma^2}}\mathbb{E}e^{\frac{\langle X-\mu,x\rangle^2}{4\sigma^2}} \\
&\leq \sqrt{2}e^{-\frac{\alpha^2}{4\sigma^2}}.
\end{aligned}$$

Finally, by a union bound over the standard basis vectors $x = e_1, e_2, \ldots, e_n$, we have

$$\mathrm{P}\Big(\|X-\mu\|_\infty \leq \alpha\Big) \geq 1 - \sqrt{2}ne^{-\frac{\alpha^2}{4\sigma^2}}.$$

□

Lemma 5.7.3. *Let $X_1, X_2, \ldots, X_N$ be i.i.d. random vectors drawn from $\mathcal{N}(0, I_m)$. Let $N \geq 10$ and $P := P_{X_N^\perp} \ldots P_{X_2^\perp} P_{X_1^\perp} \in \mathbb{R}^{m\times m}$. Then*

$$\mathrm{P}\Big(\|P\|_2^2 \leq 4\big(1-\frac{c}{m}\big)^{\lfloor\frac{N}{5}\rfloor}\Big) \geq 1 - 5^m e^{-\frac{N}{5}} \tag{5.64}$$

where $c > 0$ is an absolute constant.

Proof. This proof is based on an ε-net argument. By the definition of $\|P\|_2$, we need to bound $\|Pz\|_2$ for all vectors $z \in \mathbb{S}^{m-1}$. To this end, we will cover the unit sphere using small balls with radius ε, establish tight control of $\|Pz\|_2$ for every fixed vector z from the net, and finally take a union bound over all vectors in the net.

We first set up an ε-net. Choosing $\varepsilon = \frac{1}{2}$, according to Corollary 4.2.13 in [34], we can

find an ε-net $\mathscr{D} \subseteq \mathbb{S}^{m-1}$ such that

$$\mathbb{S}^{m-1} \subseteq \bigcup_{z \in \mathscr{D}} B(z, \varepsilon) \quad \text{and} \quad |\mathscr{D}| \leq \Big(1 + \frac{2}{\varepsilon}\Big)^m = 5^m. \tag{5.65}$$

Here, $B(z, \varepsilon)$ represents the closed ball centered at z and with radius ε, and $|\mathscr{D}|$ is the cardinality of $\mathscr{D}$. Moreover, we have (see Lemma 4.4.1 in [34])

$$\|P\|_2 \leq \frac{1}{1-\varepsilon} \max_{z \in \mathscr{D}} \|Pz\|_2 = 2 \max_{z \in \mathscr{D}} \|Pz\|_2. \tag{5.66}$$

Next, let $\beta \geq 1$, $\gamma > 0$, and $z \in \mathbb{S}^{m-1}$. Applying (5.7) and setting $\xi \sim \mathscr{N}(0, I_m)$, for $1 \leq j \leq N$, we obtain

$$\begin{aligned}
\mathrm{P}\Big(\|P_{X_j^\perp}(z)\|_2^2 \geq 1 - \gamma\Big) &= \mathrm{P}\Big(\|P_{X_j}(z)\|_2^2 \leq \gamma\Big) \\
&= \mathrm{P}\Big(\Big\langle \frac{X_j}{\|X_j\|_2}, z \Big\rangle^2 \leq \gamma\Big) \\
&= \mathrm{P}\Big(\Big\langle \frac{\xi}{\|\xi\|_2}, z \Big\rangle^2 \leq \gamma\Big).
\end{aligned}$$

By rotation invariance of the normal distribution, we may assume without loss of generality that $z = e_1 := (1, 0, \ldots, 0) \in \mathbb{R}^m$. It follows that

$$\begin{aligned}
\mathrm{P}\Big(\|P_{X_j^\perp}(z)\|_2^2 \geq 1 - \gamma\Big) &= \mathrm{P}\Big(\frac{\xi_1^2}{\|\xi\|_2^2} \leq \gamma\Big) \\
&= \mathrm{P}\Big(\frac{\xi_1^2}{\|\xi\|_2^2} \leq \gamma,\, \|\xi\|_2^2 \leq \beta m\Big) + \mathrm{P}\Big(\frac{\xi_1^2}{\|\xi\|_2^2} \leq \gamma,\, \|\xi\|_2^2 > \beta m\Big) \\
&\leq \mathrm{P}(\xi_1^2 \leq \beta\gamma m) + \mathrm{P}(\|\xi\|_2^2 \geq \beta m) \\
&\leq \sqrt{\frac{2\beta\gamma m}{\pi}} + 2\exp(-c' m(\sqrt{\beta} - 1)^2).
\end{aligned} \tag{5.67}$$

In the last step, we controlled the probability via

$$\mathrm{P}(\xi_1^2 \le \beta\gamma m) = \int_{-\sqrt{\beta\gamma m}}^{\sqrt{\beta\gamma m}} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}\,dx \le \frac{1}{\sqrt{2\pi}} \int_{-\sqrt{\beta\gamma m}}^{\sqrt{\beta\gamma m}} 1\,dx = \sqrt{\frac{2\beta\gamma m}{\pi}},$$

and used the concentration of the norm (see Theorem 3.1.1 in [34]):

$$\mathrm{P}(\|\xi\|_2^2 \ge \beta m) \le 2\exp(-c'm(\sqrt{\beta}-1)^2), \quad \beta \ge 1,$$

where $c' > 0$ is an absolute constant. In (5.67), picking $\beta = (\sqrt{\frac{3}{c'}}+1)^2$ and $\gamma = \frac{1}{12\beta m} = \frac{c}{m}$ with $c := \frac{1}{12}(\sqrt{\frac{3}{c'}}+1)^{-2}$, we have that

$$\tau := \mathrm{P}\Big(\|P_{X_j^\perp}(z)\|_2^2 \le 1 - \frac{c}{m}\Big) \ge 1 - \sqrt{\frac{1}{6\pi}} - 2e^{-3m} \ge 1 - \sqrt{\frac{1}{6\pi}} - 2e^{-3} \ge \frac{2}{3} \tag{5.68}$$

holds for all $1 \le j \le N$ and $z \in \mathbb{S}^{m-1}$. So each orthogonal projection $P_{X_j^\perp}$ can reduce the squared norm of a vector to at most $1 - \frac{c}{m}$ ratio with probability τ. Fix $z \in \mathscr{D}$. Since $X_1, X_2, \ldots, X_n$ are independent, we have

$$\begin{aligned}
\mathrm{P}\Big(\|Pz\|_2^2 \ge \Big(1-\frac{c}{m}\Big)^{\lfloor \frac{N}{5} \rfloor}\Big) &\le \sum_{k=0}^{\lfloor \frac{N}{5} \rfloor} \binom{N}{k} \tau^k (1-\tau)^{N-k} \\
&\le \sum_{k=0}^{\lfloor \frac{N}{5} \rfloor} \binom{N}{k} (1-\tau)^{N-k} && (\text{since } \tau \le 1) \\
&\le (1-\tau)^{N-\lfloor \frac{N}{5} \rfloor} \sum_{k=0}^{\lfloor \frac{N}{5} \rfloor} \binom{N}{k} \\
&\le \Big(\frac{1}{3}\Big)^{N-\lfloor \frac{N}{5} \rfloor} \sum_{k=0}^{\lfloor \frac{N}{5} \rfloor} \binom{N}{k} && (\text{by } (5.68)) \\
&\le \Big(\frac{1}{3}\Big)^{N-\lfloor \frac{N}{5} \rfloor} \Big(\frac{eN}{\lfloor \frac{N}{5} \rfloor}\Big)^{\lfloor \frac{N}{5} \rfloor} && (\text{due to } \sum_{k=0}^{l} \binom{n}{k} \le \Big(\frac{en}{l}\Big)^l). \quad (5.69)
\end{aligned}$$

Since $\frac{N}{5} - 1 < \lfloor \frac{N}{5} \rfloor \leq \frac{N}{5}$ and $N \geq 10$, we have

$$\Big(\frac{1}{3}\Big)^{N-\lfloor \frac{N}{5} \rfloor} \Big(\frac{eN}{\lfloor \frac{N}{5} \rfloor}\Big)^{\lfloor \frac{N}{5} \rfloor} \leq \Big(\frac{1}{3}\Big)^{\frac{4N}{5}} \Big(\frac{eN}{\frac{N}{5}-1}\Big)^{\frac{N}{5}} = \Big(\frac{1}{81} \cdot \frac{5e}{1-\frac{5}{N}}\Big)^{\frac{N}{5}} \leq \Big(\frac{10e}{81}\Big)^{\frac{N}{5}} \leq e^{-\frac{N}{5}}.$$

Plugging this into (5.69), we deduce that

$$\mathrm{P}\Big(\|Pz\|_2^2 \leq \Big(1 - \frac{c}{m}\Big)^{\lfloor \frac{N}{5} \rfloor}\Big) \geq 1 - e^{-\frac{N}{5}}.$$

holds for all $z \in \mathscr{D}$. By a union bound over $|\mathscr{D}| \leq 5^m$ points, we obtain

$$\mathrm{P}\Big(\max_{z \in \mathscr{D}} \|Pz\|_2^2 \leq \Big(1 - \frac{c}{m}\Big)^{\lfloor \frac{N}{5} \rfloor}\Big) \geq 1 - 5^m e^{-\frac{N}{5}}. \tag{5.70}$$

Then (5.64) follows immediately from (5.66) and (5.70). □

Moreover, we present the following result on concentration of (Gaussian) measure inequality for Lipschitz functions, which will be used in the proofs later.

Lemma 5.7.4. *Consider an n-dimensional random vector $X \sim \mathscr{N}(0, I)$ and a Lipschitz function $f : \mathbb{R}^n \to \mathbb{R}$ with Lipschitz constant $L_f > 0$, that is $|f(x) - f(y)| \leq L_f \|x - y\|_2$ for all $x, y \in \mathbb{R}^n$. Then, for all $\alpha \geq 0$,*

$$\mathrm{P}(|f(X) - \mathbb{E}f(X)| \geq \alpha) \leq 2\exp\Big(-\frac{\alpha^2}{2L_f^2}\Big).$$

A proof of Lemma 5.7.4 can be found in chapter 8 of [13]. Further, the following result provides a lower bound for the expected activation of Gaussian distribution.

Proposition 5.7.5. *Let $\rho(x) := \max\{0, x\}$ be the ReLU activation function, acting elementwise, and let $X \sim \mathscr{N}(0, \Sigma)$. Then*

$$\mathbb{E}\|\rho(X)\|_2 \geq \sqrt{\frac{\mathrm{tr}(\Sigma)}{2\pi}}.$$

To start the proof of Proposition 5.7.5, we need the following two lemmas. While these results are likely to be known, we could not find proofs in the literature so we include the

argument for completeness.

Lemma 5.7.6. *Let $\mathscr{S}$ denote the convex set of all positive semidefinite matrices A in $\mathbb{R}^{n\times n}$ with* $\mathrm{tr}(A) = 1$. *Then the extreme points of $\mathscr{S}$ are exactly the rank-*1 *matrices of the form $uu^\top$ where u is a unit vector in $\mathbb{R}^n$.*

Proof. We first let $A \in \mathscr{S}$ be an extreme point of $\mathscr{S}$ and assume $\mathrm{rank}(A) = r > 1$. Since A is positive semidefinite, the spectral decomposition of A yields $A = \sum_{i=1}^{r} \lambda_i u_i u_i^\top$ where $\lambda_i > 0$ and $\|u_i\|_2 = 1$ for $1 \le i \le r$. Then A can be rewritten as

$$A = \Big(\sum_{j=1}^{r-1} \lambda_j\Big) B + \lambda_r u_r u_r^\top$$

where $B = \sum_{i=1}^{r-1} \frac{\lambda_i}{\sum_{j=1}^{r-1}\lambda_j} u_i u_i^\top$. Note that B and $u_r u_r^\top$ are distinct positive semidefinite matrices with $\mathrm{tr}(B) = \mathrm{tr}(u_r u_r^\top) = 1$, and $\sum_{j=1}^{r} \lambda_j = \mathrm{tr}(A) = 1$. Thus, $B, u_r u_r^\top \in \mathscr{S}$ and A is in the open line segment joining B and $u_r u_r^\top$, which is a contradiction. So any extreme point of $\mathscr{S}$ is a rank-1 matrix of the form $A = uu^\top$ with $\|u\|_2 = 1$.

Conversely, consider any rank-1 matrix $A = uu^\top$ with $\|u\|_2 = 1$. Then we have $A \in \mathscr{S}$. Assume that A lies in an open segment in $\mathscr{S}$ connecting two distinct matrices $A_1, A_2 \in \mathscr{S}$, that is

$$A = \alpha_1 A_1 + \alpha_2 A_2 \tag{5.71}$$

where $\alpha_1 + \alpha_2 = 1$ and $0 < \alpha_1 \le \alpha_2$. Additionally, for any $x \in \ker(A)$, we have

$$0 = x^\top A x = \alpha_1 x^\top A_1 x + \alpha_2 x^\top A_2 x \tag{5.72}$$

and thus $A_1 x = A_2 x = 0$. It implies $\ker(A) \subseteq \ker(A_1) \cap \ker(A_2)$. By the rank–nullity theorem, we get $1 = \mathrm{rank}(A) \ge \max\{\mathrm{rank}(A_1), \mathrm{rank}(A_2)\}$. Since A_1 and A_2 are distinct matrices in $\mathscr{S}$, we have $\mathrm{rank}(A_1) = \mathrm{rank}(A_2) = 1$ and there exist unit vectors u_1, u_2 such that $A_1 = u_1 u_1^\top$, $A_2 = u_2 u_2^\top$,

and $u_1 \neq \pm u_2$. Hence,

$$\operatorname{rank}(A_1 + A_2) = \operatorname{rank}([u_1, u_2][u_1, u_2]^\top) = \operatorname{rank}([u_1, u_2]) = 2.$$

Moreover, it follows from (5.71) that $A = \alpha_1(A_1 + A_2) + (\alpha_2 - \alpha_1)A_2$. Due to $\alpha_2 - \alpha_1 \geq 0$, one can get $\operatorname{rank}(A) \geq \operatorname{rank}(A_1 + A_2) = 2$ by a similar argument we applied in (5.72). However, this contradicts the assumption that A is a rank-1 matrix. Therefore, for any unit vector u, $A = uu^\top$ is an extreme point of S. □

Lemma 5.7.7. *Suppose* $X \sim \mathcal{N}(0, \Sigma)$. *Then* $\mathbb{E}\|X\|_2 \geq \sqrt{\frac{2\operatorname{tr}(\Sigma)}{\pi}}$.

Proof. Without loss of generality, we can assume that $\operatorname{tr}(\Sigma) = 1$. Let $Z \sim \mathcal{N}(0, I)$. Since $\Sigma^{\frac{1}{2}} Z \sim \mathcal{N}(0, \Sigma)$, we have

$$\mathbb{E}\|X\|_2 = \mathbb{E}\|\Sigma^{\frac{1}{2}} Z\|_2 = \mathbb{E}\sqrt{Z^\top \Sigma Z}. \tag{5.73}$$

Define a function $f(A) := \mathbb{E}\sqrt{Z^\top A Z}$ and let $\mathscr{S}$ denote the set of all positive semidefinite matrices whose traces are equal to 1. Then $f(A)$ is continuous and concave over $\mathscr{S}$ that is convex and compact. By Bauer maximum principle, $f(A)$ attains its minimum at some extreme point $\widetilde{A}$ of $\mathscr{S}$. According to Lemma 5.7.6, $\widetilde{A} = uu^\top$ with $\|u\|_2 = 1$. If follows that

$$\min_{A \in \mathscr{S}} f(A) = f(\widetilde{A}) = \mathbb{E}\sqrt{Z^\top \widetilde{A} Z} = \mathbb{E}|u^\top Z| = \sqrt{\frac{2}{\pi}}. \tag{5.74}$$

In the last step, we used the fact $u^\top Z \sim \mathcal{N}(0, 1)$. Combining (5.73) and (5.74), we obtain

$$\mathbb{E}\|X\|_2 = f(\Sigma) \geq \min_{A \in \mathscr{S}} f(A) = \sqrt{\frac{2}{\pi}}.$$

This completes the proof. □

Lemma 5.7.8. *Given an n-dimensional random vector* $X \sim \mathcal{N}(0, \Sigma)$, *we have*

$$\mathbb{E}\|\rho(X)\|_2 \geq \frac{1}{2}\mathbb{E}\|X\|_2$$

where $\rho(x) = \max\{0, x\}$ *is the ReLU activation function.*

Proof. We divide $\mathbb{R}^n$ into $J := 2^{n-1}$ pairs of orthants $\{(A_i, B_i)\}_{i=1}^{J}$ such that $-A_i = B_i$. For example, $\{(x_1, x_2, \ldots, x_n) : x_i > 0, i = 1, 2, \ldots, n\}$ and $\{(x_1, x_2, \ldots, x_n) : x_i < 0, i = 1, 2, \ldots, n\}$ compose one of these pairs. Since X is symmetric, that is, X and $-X$ have the same distribution, one can get

$$\int_{A_i} \|\rho(-x)\|_2 \, dP_X = \int_{B_i} \|\rho(x)\|_2 \, dP_X \tag{5.75}$$

and

$$\int_{A_i} \|x\|_2 \, dP_X = \int_{B_i} \|x\|_2 \, dP_X \tag{5.76}$$

where P_X denotes the probability distribution of X. It follows that

$$\begin{aligned}
\mathbb{E}\|\rho(X)\|_2 &= \int_{\mathbb{R}^n} \|\rho(x)\|_2 \, dP_X \\
&= \sum_{j=1}^{J} \int_{A_j \cup B_j} \|\rho(x)\|_2 \, dP_X \\
&= \sum_{j=1}^{J} \int_{A_j} \|\rho(x)\|_2 \, dP_X + \int_{A_j} \|\rho(-x)\|_2 \, dP_X && \text{(using (5.75))} \\
&\geq \sum_{j=1}^{J} \int_{A_j} \|\rho(x) + \rho(-x)\|_2 \, dP_X && \text{(by triangle inequality)} \\
&= \sum_{j=1}^{J} \int_{A_j} \|x\|_2 \, dP_X \\
&= \frac{1}{2} \sum_{j=1}^{J} \int_{A_j \cup B_j} \|x\|_2 \, dP_X && \text{(using (5.76))} \\
&= \frac{1}{2} \mathbb{E}\|X\|_2.
\end{aligned}$$

□

Proposition 5.7.5 then follows immediately from Lemma 5.7.7 and Lemma 5.7.8.

5.8 Perturbation analysis for underdetermined systems

In this section, we investigate the minimal ℓ_∞ norm solutions of perturbed underdetermined linear systems like (5.22), which can be used to bound the ℓ_∞ norm of $\widetilde{w}$ generated by the perfect data alignment. Specifically, consider a matrix $X \in \mathbb{R}^{m\times N}$ with $\operatorname{rank}(X) = m < N$. It admits the singular value decomposition

$$X = USV^\top \tag{5.77}$$

where $U = [U_1, \ldots, U_m] \in \mathbb{R}^{m\times m}$, $V = [V_1, \ldots, V_m] \in \mathbb{R}^{N\times m}$ have orthonormal columns, and $S = \operatorname{diag}(\sigma_1, \ldots, \sigma_m)$ consists of singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_m > 0$. Moreover, suppose $\varepsilon > 0$, $w \in \mathbb{R}^N$, and $E \in \mathbb{R}^{m\times N}$ satisfying $\|E\|_2 \leq \varepsilon \|X\|_2$. Let $\widetilde{X} := X + E$ be the perturbed matrix and define

$$\hat{w} := \arg\min \|z\|_\infty \text{ subject to } Xz = Xw, \tag{5.78}$$

$$\widetilde{w} := \arg\min \|z\|_\infty \text{ subject to } \widetilde{X}z = Xw. \tag{5.79}$$

Our goal is to evaluate the ratio $\frac{\|\widetilde{w}\|_\infty}{\|\hat{w}\|_\infty}$.

The proposition below highlights the fact that one can construct systems where arbitrarily small perturbations can yield arbitrarily divergent solutions. The proof relies on the system being ill-conditioned, and on a particular construction of X and E to exploit the ill-conditioning.

Proposition 5.8.1. *For $\varepsilon, \gamma \in (0,1)$, there exist a matrix $X \in \mathbb{R}^{m\times N}$, a perturbed version $\widetilde{X} = X + E$ with $\|E\|_2 \leq \varepsilon\|X\|_2$, and a unit vector $w \in \mathbb{R}^N$, so that the optimal solutions to* (5.78) *and* (5.79) *satisfy $\frac{\|\widetilde{w}\|_\infty}{\|\hat{w}\|_\infty} = \frac{1}{\gamma}$.*

Proof. Let $U \in \mathbb{R}^{m\times m}$ be any orthogonal matrix and let $V \in \mathbb{R}^{N\times m}$ be the first m columns of a normalized Hadamard matrix of order N. Then we have $V^\top V = I$ and entries of V are either $\frac{1}{\sqrt{N}}$ or $-\frac{1}{\sqrt{N}}$. Set $X = USV^\top$ where $S \in \mathbb{R}^{m\times m}$ is diagonal with diagonal elements

$\sigma_1 = \sigma_2 = \ldots = \sigma_{m-1} = 1$ and $\sigma_m = \varepsilon$. Define a rank one matrix $E = \varepsilon(\gamma - 1)U_m V_m^\top$. Then we have

$$\frac{\|E\|_2}{\|X\|_2} = \varepsilon(1-\gamma) < \varepsilon, \quad \widetilde{X} = X + E = U\mathrm{diag}(1,\ldots,1,\varepsilon\gamma)V^\top.$$

Picking a unit vector $w = \varepsilon V S^{-1} e_m$ with $e_m := (0,\ldots,0,1) \in \mathbb{R}^m$, the feasibility condition in (5.78), together with the definition of X, implies that $Xz = Xw$ is equivalent to

$$V^\top z = e_m. \tag{5.80}$$

Since $VV^\top z = P_{\mathrm{Im}(V)}(z)$ is the orthogonal projection of z onto the image of V, for any feasible z satisfying (5.80), we have

$$\|z\|_\infty \geq \frac{\|z\|_2}{\sqrt{N}} \geq \frac{\|VV^\top z\|_2}{\sqrt{N}} = \frac{\|V^\top z\|_2}{\sqrt{N}} = \frac{\|e_m\|_2}{\sqrt{N}} = \frac{1}{\sqrt{N}}.$$

Note that $z = V_m$ satisfies (5.80) and $\|V_m\|_\infty = \frac{1}{\sqrt{N}}$ achieves the lower bound. Thus, we have found an optimal solution $\hat{w} = V_m$ with $\|\hat{w}\|_\infty = \frac{1}{\sqrt{N}}$.

Meanwhile the corresponding feasibility condition in (5.79), coupled with the definition of $\widetilde{X}$, implies that $\widetilde{X}z = Xw$ can be rewritten as $V^\top z = \frac{1}{\gamma} e_m$. By a similar argument we used for solving (5.78), we obtain that $\widetilde{w} = \frac{1}{\gamma} V_m$ is an optimal solution to (5.79) and thus $\|\widetilde{w}\|_\infty = \frac{1}{\gamma\sqrt{N}}$. Therefore, we have $\frac{\|\widetilde{w}\|_\infty}{\|\hat{w}\|_\infty} = \frac{1}{\gamma}$ as desired. □

Proposition 5.8.1 constructs a scenario in which adjusting the weights to achieve $\widetilde{X}\widetilde{w} = X\hat{w} = Xw$, under even a small perturbation of X, inexorably leads to a large increase in the infinity norm of $\widetilde{w}$. In Proposition 5.8.3, we consider a more reasonable scenario where the original weights w is Gaussian that is more likely to be representative of ones encountered in practice. The proof of the following lemma follows [1].

Lemma 5.8.2. *Let $\|\cdot\|$ be any vector norm on $\mathbb{R}^n$. Let $X \sim \mathcal{N}(0,\Sigma_1)$ and $Y \sim \mathcal{N}(0,\Sigma_2)$ be*

n-dimensional random vectors. Suppose $\Sigma_1 \preceq \Sigma_2$. *Then, for* $t \geq 0$, *we have*

$$\mathrm{P}(\|X\| \leq t) \geq \mathrm{P}(\|Y\| \leq t).$$

Proof. Fix $t \geq 0$. Define $g : \mathbb{R}^n \to [0,1]$ by

$$g(z) := \mathrm{P}(\|X+z\| \leq t) = \int_{\mathbb{R}^n} f_X(x) \mathbb{1}_{\{\|x+z\| \leq t\}} \, dx$$

where

$$f_X(x) := (2\pi)^{-\frac{n}{2}} \det(\Sigma_1)^{-\frac{1}{2}} \exp(-\frac{1}{2} x^\top \Sigma_1^{-1} x)$$

is the density function of X. Since $\log f_X(x) = -\frac{1}{2} x^\top \Sigma_1^{-1} x$ is concave and $\mathbb{1}_{\{\|x+z\| \leq t\}}$ is an indicator function of a convex set, both $f_X(x)$ and $\mathbb{1}_{\{\|x+z\| \leq t\}}$ are log-concave. It follows that the product $h(x,z) := f_X(x) \mathbb{1}_{\{\|x+z\| \leq t\}}$ is also log-concave. Applying the Prékopa–Leindler inequality [29, 30], the marginalization $g(z) = \int_{\mathbb{R}^n} h(x,z) \, dx$ preserves log-concavity. Additionally, by change of variables and the symmetry of $f_X(x)$, we have

$$g(-z) = \int_{\mathbb{R}^n} f_X(x) \mathbb{1}_{\{\|x-z\| \leq t\}} \, dx = \int_{\mathbb{R}^n} f_X(x) \mathbb{1}_{\{\|x+z\| \leq t\}} \, dx = g(z).$$

So $g(z)$ is a log-concave even function, which implies that, for any $z \in \mathbb{R}^n$,

$$g(z) = g(z)^{\frac{1}{2}} g(-z)^{\frac{1}{2}} \leq g\Big(\frac{1}{2} z - \frac{1}{2} z\Big) = g(0) = \mathrm{P}(\|X\| \leq t). \tag{5.81}$$

Now, let $Z \sim \mathcal{N}(0, \Sigma_2 - \Sigma_1)$ be independent of X. Then $X + Z \overset{d}{=} Y \sim \mathcal{N}(0, \Sigma_2)$ and, by (5.81),

$\mathbb{E}g(Z) \leq \mathrm{P}(\|X\| \leq t)$. It follows that

$$\begin{aligned}
\mathrm{P}(\|X\| \leq t) &\geq \mathbb{E}g(Z) \\
&= \int_{\mathbb{R}^n} f_Z(z) g(z)\, dz \\
&= \int_{\mathbb{R}^n} \int_{\mathbb{R}^n} f_X(x) f_Z(z) \mathbb{1}_{\{\|x+z\| \leq t\}}\, dx\, dz \\
&= \int_{\mathbb{R}^n \times \mathbb{R}^n} f_{(X,Z)}(x,z) \mathbb{1}_{\{\|x+z\| \leq t\}}\, d(x,z) \\
&= \mathrm{P}(\|X+Z\| \leq t) \\
&= \mathrm{P}(\|Y\| \leq t)
\end{aligned}$$

where $f_Z(z)$ and $f_{(X,Z)}(x,z)$ are density functions of Z and (X,Z) respectively. □

Proposition 5.8.3. *Let $X \in \mathbb{R}^{m\times N}$ admit the singular value decomposition $X = USV^\top$ as in* (5.77) *and let $w \in \mathbb{R}^N$ be a random vector with i.i.d. $\mathcal{N}(0,1)$ entries. Let $p \in \mathbb{N}$ with $p \geq 2$. Given $\varepsilon \in (0,1)$, suppose $\widetilde{X} = X + E \in \mathbb{R}^{m\times N}$ with $\|E\|_2 \leq \varepsilon\sigma_1 < \sigma_m$.Then, with probability at least $1 - \frac{2}{N^{p-1}}$,*

$$\|\widetilde{w}\|_\infty \leq \frac{\sigma_1}{\sigma_m - \varepsilon\sigma_1}\sqrt{2p\log N}$$

holds for all optimal solutions $\widetilde{w}$ of (5.79).

Proof. Let $w^\sharp := VV^\top w$ be the orthogonal projection of w onto $\mathrm{Im}(V)$. Let $\widetilde{V} = [V, \hat{V}] \in \mathbb{R}^{N\times N}$ be an expansion of V such that $\widetilde{V}$ is orthogonal. Define

$$\widetilde{\mathscr{E}} := U^\top E\widetilde{V} = [U^\top EV, U^\top E\hat{V}] = [\mathscr{E}, \hat{\mathscr{E}}] \in \mathbb{R}^{m\times N}$$

where $\mathscr{E} := U^\top EV$ and $\hat{\mathscr{E}} := U^\top E\hat{V}$. Then $E = U\widetilde{\mathscr{E}}\widetilde{V}^\top$ and thus

$$\varepsilon\sigma_1 \geq \|E\|_2 = \|\widetilde{\mathscr{E}}\|_2 \geq \|\mathscr{E}\|_2. \tag{5.82}$$

Define $z^\sharp := V(S+\mathscr{E})^{-1}SV^\top w \in \mathbb{R}^N$. Since $\widetilde{\mathscr{E}}\widetilde{V}^\top V = \mathscr{E}$, we have

$$\begin{aligned}\widetilde{X}z^\sharp &= Xz^\sharp + Ez^\sharp \\ &= US(S+\mathscr{E})^{-1}SV^\top w + U\widetilde{\mathscr{E}}\widetilde{V}^\top V(S+\mathscr{E})^{-1}SV^\top w \\ &= US(S+\mathscr{E})^{-1}SV^\top w + U\mathscr{E}(S+\mathscr{E})^{-1}SV^\top w \\ &= USV^\top w \\ &= Xw.\end{aligned}$$

Moreover, since $w \sim \mathscr{N}(0,I)$, we have $z^\sharp \sim \mathscr{N}(0,BB^\top)$ with $B := V(S+\mathscr{E})^{-1}S$ and thus

$$BB^\top \preccurlyeq \|BB^\top\|_2 I = \|B\|_2^2 I = \|(S+\mathscr{E})^{-1}S\|_2^2 I \preccurlyeq \Big(\frac{\sigma_1}{\sigma_m - \|\mathscr{E}\|_2}\Big)^2 I \preccurlyeq \Big(\frac{\sigma_1}{\sigma_m - \varepsilon\sigma_1}\Big)^2 I. \quad (5.83)$$

Applying Lemma 5.8.2 to (5.83) with $\Sigma_1 = BB^\top$ and $\Sigma_2 = (\frac{\sigma_1}{\sigma_m-\varepsilon\sigma_1})^2 I$, we obtain that, for $t \geq 0$,

$$\mathrm{P}(\|z^\sharp\|_\infty \leq t) \geq \mathrm{P}\Big(\Big\|\frac{\sigma_1\xi}{\sigma_m-\varepsilon\sigma_1}\Big\|_\infty \leq t\Big) \geq 1 - 2N\exp\Big(-\frac{1}{2}\big(\frac{\sigma_m-\varepsilon\sigma_1}{\sigma_1}\big)^2 t^2\Big) \quad (5.84)$$

where $\xi \sim \mathscr{N}(0,I)$. In the last inequality, we used the following concentration inequality

$$\mathrm{P}(\|\xi\|_\infty \leq t) \geq 1 - 2Ne^{-\frac{t^2}{2}}, \quad t \geq 0.$$

Choosing $t = \frac{\sigma_1}{\sigma_m-\varepsilon\sigma_1}\sqrt{2p\log N}$ in (5.84), we obtain

$$\mathrm{P}\Big(\|z^\sharp\|_\infty \leq \frac{\sigma_1}{\sigma_m-\varepsilon\sigma_1}\sqrt{2p\log N}\Big) \geq 1 - \frac{2}{N^{p-1}}.$$

Further, since $z^\sharp$ is a feasible vector of (5.79), we have $\|\widetilde{w}\|_\infty \leq \|z^\sharp\|_\infty$. Therefore, with probability at least $1 - \frac{2}{N^{p-1}}$,

$$\|\widetilde{w}\|_\infty \leq \frac{\sigma_1}{\sigma_m-\varepsilon\sigma_1}\sqrt{2p\log N}.$$

□

5.9 Acknowledgements

The authors thank Yixuan Zhou for discussions on the numerical experiments in this paper. This work was supported in part by National Science Foundation Grant DMS-2012546 and a Simons Fellowship. This chapter, in full, is joint work with Rayan Saab, and is currently being prepared for submission for publication. The dissertation author was the primary investigator and author of this material.

References

[1] Iosif Pinelis (https://mathoverflow.net/users/36721/iosif-pinelis). *ℓ_∞ norm of two gaussian vector*. MathOverflow. URL:https://mathoverflow.net/q/410242. 2021. eprint: https://mathoverflow.net/q/410242.

[2] Nabih N Abdelmalek. "Minimum L_∞ solution of underdetermined systems of linear equations". In: *Journal of Approximation Theory* 20.1 (1977), pp. 57–69.

[3] Ryan Alweiss, Yang P Liu, and Mehtaab Sawhney. "Discrepancy minimization via a self-balancing walk". In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. 2021, pp. 14–20.

[4] James A Cadzow. "A finite algorithm for the minimum L_∞ solution to a system of consistent linear equations". In: *SIAM Journal on Numerical Analysis* 10.4 (1973), pp. 607–617.

[5] James A Cadzow. "An Efficient Algorithmic Procedure for Obtaining a Minimum L_∞-Norm Solution to a System of Consistent Linear Equations". In: *SIAM Journal on Numerical Analysis* 11.6 (1974), pp. 1151–1165.

[6] Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. “Zeroq: A novel zero shot quantization framework”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 13169–13178.

[7] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. “A survey of model compression and acceleration for deep neural networks”. In: *arXiv preprint arXiv:1710.09282* (2017).

[8] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. “Pact: Parameterized clipping activation for quantized neural networks”. In: *arXiv preprint arXiv:1805.06085* (2018).

[9] Yoni Choukroun, Eli Kravchik, Fan Yang, and Pavel Kisilev. “Low-bit Quantization of Neural Networks for Efficient Inference.” In: *ICCV Workshops*. 2019, pp. 3009–3018.

[10] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. “Binaryconnect: Training deep neural networks with binary weights during propagations”. In: *Advances in neural information processing systems*. 2015, pp. 3123–3131.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[12] Lei Deng, Guoqi Li, Song Han, Luping Shi, and Yuan Xie. “Model compression and hardware acceleration for neural networks: A comprehensive survey”. In: *Proceedings of the IEEE* 108.4 (2020), pp. 485–532.

[13] Simon Foucart and Holger Rauhut. “An invitation to compressive sensing”. In: *A mathematical introduction to compressive sensing*. Springer, 2013, pp. 1–39.

[14] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. “Gptq: Accurate post-training quantization for generative pre-trained transformers”. In: *arXiv preprint arXiv:2210.17323* (2022).

[15] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. “A survey of quantization methods for efficient neural network inference”. In: *arXiv preprint arXiv:2103.13630* (2021).

[16] Yunhui Guo. “A survey on methods and theories of quantized neural networks”. In: *arXiv preprint arXiv:1808.04752* (2018).

[17] Josef Hadar and William R Russell. “Rules for ordering uncertain prospects”. In: *The American economic review* 59.1 (1969), pp. 25–34.

[18] Giora Hanoch and Haim Levy. “The efficiency analysis of choices involving risk”. In: *Stochastic Optimization Models in Finance*. Elsevier, 1975, pp. 89–100.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[20] Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. “Improving post training neural quantization: Layer-wise calibration and integer programming”. In: *arXiv preprint arXiv:2006.10518* (2020).

[21] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. “Quantization and training of neural networks for efficient integer-arithmetic-only inference”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2704–2713.

[22] Raghuraman Krishnamoorthi. “Quantizing deep convolutional networks for efficient inference: A whitepaper”. In: *arXiv preprint arXiv:1806.08342* (2018).

[23] Eric Lybrand and Rayan Saab. “A Greedy Algorithm for Quantizing Neural Networks”. In: *Journal of Machine Learning Research* 22.156 (2021), pp. 1–38.

[24] Mark Machina and John Pratt. “Increasing risk: some direct constructions”. In: *Journal of Risk and Uncertainty* 14.2 (1997), pp. 103–127.

[25] Johannes Maly and Rayan Saab. “A simple approach for quantizing neural networks”. In: *Applied and Computational Harmonic Analysis* 66 (2023), pp. 138–150.

[26] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen B. “Up or down? adaptive rounding for post-training quantization”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 7197–7206.

[27] James O’ Neill. “An overview of neural network compression”. In: *arXiv preprint arXiv:2006.03669* (2020).

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. “Pytorch: An imperative style, high-performance deep learning library”. In: *Advances in neural information processing systems* 32 (2019), pp. 8026–8037.

[29] András Prékopa. “Logarithmic concave measures with application to stochastic programming”. In: *Acta Scientiarum Mathematicarum* 32 (1971), pp. 301–316.

[30] András Prékopa. “On logarithmic concave measures and functions”. In: *Acta Scientiarum Mathematicarum* 34 (1973), pp. 335–343.

[31] Michael Rothschild and Joseph E Stiglitz. “Increasing risk: I. A definition”. In: *Journal of Economic theory* 2.3 (1970), pp. 225–243.

[32] Moshe Shaked and J George Shanthikumar. *Stochastic orders*. Springer, 2007.

[33] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *International Conference on Learning Representations* (2015).

[34] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*. Vol. 47. Cambridge university press, 2018.

[35] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. “Haq: Hardware-aware automated quantization with mixed precision”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8612–8620.

[36] Peisong Wang, Qiang Chen, Xiangyu He, and Jian Cheng. “Towards accurate post-training network quantization via bit-split and stitching”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9847–9856.

[37] Dongqing Zhang, Jiaolong Yang, Dongqiangzi Ye, and Gang Hua. “Lq-nets: Learned quantization for highly accurate and compact deep neural networks”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 365–382.

[38] Jinjie Zhang, Yixuan Zhou, and Rayan Saab. “Post-training quantization for neural networks with provable guarantees”. In: *SIAM Journal on Mathematics of Data Science* 5.2 (2023), pp. 373–399.

[39] Ritchie Zhao, Yuwei Hu, Jordan Dotzel, Chris De Sa, and Zhiru Zhang. “Improving neural network quantization without retraining using outlier channel splitting”. In: *International conference on machine learning*. PMLR. 2019, pp. 7543–7552.

[40] Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen. “Incremental network quantization: Towards lossless cnns with low-precision weights”. In: *arXiv preprint arXiv:1702.03044* (2017).