

# Optimal Quantization for Matrix Multiplication

Or Ordentlich and Yury Polyanskiy

## Abstract

Recent work in machine learning community proposed multiple methods for performing lossy compression (quantization) of large matrices. This quantization is important for accelerating matrix multiplication (main component of large language models), which is often bottlenecked by the speed of loading these matrices from memory. Unlike classical vector quantization and rate-distortion theory, the goal of these new compression algorithms is to be able to approximate not the matrices themselves, but their matrix product. Specifically, given a pair of real matrices  $A, B$  an encoder (compressor) is applied to each of them independently producing descriptions with  $R$  bits per entry. These representations subsequently are used by the decoder to estimate matrix product  $A^\top B$ . In this work, we provide a non-asymptotic lower bound on the mean squared error of this approximation (as a function of rate  $R$ ) for the case of matrices  $A, B$  with iid Gaussian entries. Algorithmically, we construct a universal quantizer based on nested lattices with an explicit guarantee of approximation error for any (non-random) pair of matrices  $A, B$  in terms of only Frobenius norms  $\|A\|_F, \|B\|_F$  and  $\|A^\top B\|_F$ . For iid Gaussian matrices our quantizer achieves the lower bound and is, thus, asymptotically optimal. A practical low-complexity version of our quantizer achieves performance quite close to optimal. In information-theoretic terms we derive rate-distortion function for matrix multiplication of iid Gaussian matrices.

## CONTENTS

<b>I</b>	<b>Introduction and main results</b>	2
I-A	Importance of quantization for modern applications . . . . .	4
I-B	Sketch of the proof . . . . .	5
I-C	Related work . . . . .	6
I-D	Paper organization . . . . .	8
I-E	Notation . . . . .	9
<b>II</b>	<b>Compression for Inner-Product Computation: General Problem Setup and Simple Bounds</b>	9
II-A	Optimal Decoder and Error Expressions . . . . .	10
II-B	Simple Lower Bounds . . . . .	10
<b>III</b>	<b>Compression for Inner-Product Computation: The Symmetric Case</b>	11
III-A	Upper Bound . . . . .	11
III-B	Lower Bound . . . . .	12
III-C	The Symmetric Gaussian case . . . . .	14
<b>IV</b>	<b>Compression for Matrix Multiplication</b>	14
IV-A	Setup . . . . .	14
IV-B	Basic Properties and Bounds . . . . .	14
IV-C	Maximum Entropy Matrices . . . . .	15
IV-D	Fundamental Limits . . . . .	16
IV-E	The Symmetric Gaussian case . . . . .	17
<b>V</b>	<b>Lattice Quantization Scheme for Matrix Multiplication of Arbitrary Matrices</b>	17
V-A	Proof of Theorem 11, Part 1 . . . . .	20
V-A1	Dithered Nested Lattice Quantization for Inner Product . . . . .	20
V-A2	Analysis . . . . .	20
V-A3	Time Sharing . . . . .	22
V-B	Proof of Theorem 11, Part 2 . . . . .	23
V-C	Proof of Theorem 12 . . . . .	24

<b>VI</b>	<b>Practical Implementation of Nested Lattice Quantizers</b>	24
<b>VII</b>	<b>Open problems</b>	28
	<b>Appendix A: Convex envelope of <math>\Gamma_1(R)</math></b>	29
	<b>Appendix B: Projections of Random Uniform Orthogonal Vectors</b>	30
	<b>References</b>	31

## I. INTRODUCTION AND MAIN RESULTS

Matrix multiplication is a key component of many numerical algorithms, and is often the dominant factor in the runtime of a program. With the surge of deep neural nets (DNNs) and large language models (LLMs), finding more efficient ways to perform matrix multiplication have become one of the most pressing challenges. Classical work in this field focused on minimizing the number of required operations [1], [2], [3], [4]. Specifics of contemporary problems, however, require rethinking this classical approach to matrix multiplication. First, in machine learning applications requirements for precision of computing matrix products are quite lax. Second, modern computational hardware is often bottlenecked by the memory bandwidth. A natural solution explored by many researchers is to apply lossy compression to matrices leading to deterioration in precision but improvement in the amount of data transferred between memory and computation cores.

We formalize this problem as follows. Consider a pair of matrices  $A \in \mathbb{R}^{n \times a}$  and  $B \in \mathbb{R}^{n \times b}$  which need to be described using  $R$  bits per entry (using separate compressors), such that a decoder that obtains bit descriptions of both matrices can estimate  $\widehat{A^\top B}$ . The metric for gauging quality of approximation that we will use is the squared error between  $ab$  entries of  $\widehat{A^\top B}$  and  $A^\top B$ . Note that unlike classical vector quantization, we are requiring compression algorithms to be tailored to the special task of matrix multiplication. As a practical motivation, in Section I-A below we argue that reducing  $R$  down to a few bits/entry is necessary for LLMs to fully leverage modern matrix multiplication hardware.

Our main result shows existence of universal quantizers (based on lattices) which compress  $A$  and  $B$  to  $R$  bits/entry and come with explicit precision guarantees. Furthermore, we also show that these guarantees cannot be generally improved by proving a matching lower bound for the case of matrices  $A$  and  $B$  with iid Gaussian entries. We emphasize, though, that quantizers *are* universal and do not require Gaussian matrices.

To introduce our main results, let us define the function

$$\Gamma(R) = \begin{cases} 1 - (1 - (2 \cdot 2^{-2R^*} - 2^{-4R^*})) \frac{R}{R^*} & R < R^* \\ 2 \cdot 2^{-2R} - 2^{-4R} & R \geq R^* \end{cases} \quad (1)$$

where  $R^* \approx 0.906$  is the solution to the fixed-point equation

$$R = \frac{1}{2} \log_2(1 + 4R \ln 2) \quad (2)$$

It will turn out that  $\Gamma(R)$  is distortion-rate function for the matrix multiplication of iid Gaussian matrices.

We say that a matrix  $A \in \mathbb{R}^{n \times m}$  has “bounded entries” if  $|a_{i,j}| \in \{0\} \cup [2^{-2000}, 2^{2000}]$  for all  $i \in [n], j \in [m]$ . This extremely mild condition guarantees that we can describe the  $\ell_2$  norm of each column of  $A$  with small multiplicative error using  $o(n)$  bits (see Section V). Our first result is the following.

*Theorem 1:* For any  $\varepsilon > 0$  and sufficiently large  $n$

- 1) There exist randomized encoders  $f_1 : \mathbb{R}^{n \times a} \rightarrow [2^{naR}]$ ,  $f_2 : \mathbb{R}^{n \times b} \rightarrow [2^{nbR}]$ , and decoder  $g : [2^{naR}] \times [2^{nbR}] \rightarrow \mathbb{R}^{a \times b}$ , such that for any  $A \in \mathbb{R}^{n \times a}$  and  $B \in \mathbb{R}^{n \times b}$  with bounded entries we have

$$\mathbb{E} \|A^\top B - g(f_1(A), f_2(B))\|_F^2 < \|A^\top B\|_F^2 \cdot (\Gamma^2(R) + \varepsilon) + \frac{\|A\|_F^2 \|B\|_F^2}{n} (\Gamma(R) - \Gamma^2(R) + \varepsilon). \quad (3)$$

Furthermore, denoting  $A = [a_1 | \dots | a_a]$  and  $B = [b_1 | \dots | b_b]$ , where  $a_i, b_j \in \mathbb{R}^n$  are the columns of  $A$  and  $B$ ,  $C = A^\top B$  and  $\hat{C} = g(f_1(A), f_2(B))$ , they have that

$$\mathbb{E} (C_{i,j} - \hat{C}_{i,j})^2 \leq C_{i,j}^2 \cdot (\Gamma^2(R) + \varepsilon) + \frac{\|a_i\|^2 \|b_j\|^2}{n} (\Gamma(R) - \Gamma^2(R) + \varepsilon), \quad \forall i \in [a], j \in [b]. \quad (4)$$

- 2) There exist a randomized encoder  $f : \mathbb{R}^{n \times a} \rightarrow [2^{naR}]$  and decoder  $g : [2^{naR}] \times \mathbb{R}^{n \times b} \rightarrow \mathbb{R}^{a \times b}$ , such that for any  $A \in \mathbb{R}^{n \times a}$  and  $B \in \mathbb{R}^{n \times b}$  with bounded entries we have

$$\mathbb{E} \|A^\top B - g(f(A), B)\|_F^2 < \|A^\top B\|_F^2 \cdot (2^{-4R} + \varepsilon) + \frac{\|A\|_F^2 \|B\|_F^2}{n} (2^{-2R} - 2^{-4R} + \varepsilon). \quad (5)$$

Furthermore,

$$\mathbb{E}(C_{i,j} - \hat{C}_{i,j})^2 \leq C_{i,j}^2 \cdot (2^{-4R} + \varepsilon) + \frac{\|a_i\|^2 \|b_j\|^2}{n} (2^{-2R} - 2^{-4R} + \varepsilon), \quad \forall i \in [a], j \in [b]. \quad (6)$$

where  $\hat{C} = g(f(A), B)$ .

Note that two parts simply describe the cases, where both or only one matrix needs to be compressed.<sup>1</sup> For the second part, where only  $A$  needs to be compressed, note that if the entries of  $A$  are iid  $\mathcal{N}(0, \sigma^2)$  and  $B$  is the  $n \times n$  identity matrix, the right hand sides of (5) and (6) read  $na\sigma^2(2^{-2R} + 2\varepsilon)$  and  $\sigma^2(2^{-2R} + 2\varepsilon)$ , respectively, which are optimal, as they correspond to the Gaussian rate-distortion function. For the first part, where both matrices are compressed, our scheme operates by compressing each column of  $A$  and  $B$  using the same (randomized) quantizer  $f_{\text{col}} : \mathbb{R}^n \rightarrow [2^{nR}]$ , which is applied repeatedly to every column. The decoder  $g$  simply estimates each column to get matrices  $\hat{A}$  and  $\hat{B}$  and computes their matrix product. For rates  $R > R^*$  compressor  $f_{\text{col}}$  compresses all coordinates, while for  $R < R^*$  a fraction  $1 - (\frac{R}{R^*})$  of coordinates are ignored (mapped to 0). The ‘‘sparsification’’ is necessary to achieve asymptotically optimal quantization and was one of the main surprises of this work, compared to standard vector quantization.

To get a feel for Theorem 1 let us consider independent matrices  $A$  and  $B$  drawn iid Gaussian  $\mathcal{N}(0, \sigma^2)$ . We have that  $\mathbb{E} \|A^\top B\|_F^2 = \frac{\mathbb{E} \|A\|_F^2 \|B\|_F^2}{n} = \sigma^4 \cdot nab$  in this case and Theorem 1 shows estimate

$$\mathbb{E} \|A^\top B - \widehat{A^\top B}\|_F^2 \leq \sigma^4 nab(\Gamma(R) + \varepsilon).$$

It turns out that this is the best possible approximation (at this compression rate), as shown in our next result.

*Theorem 2:* Let  $A \in \mathbb{R}^{n \times a}$  and  $B \in \mathbb{R}^{n \times b}$  be independent random matrices, with iid  $\mathcal{N}(0, \sigma^2)$  entries.

- 1) For any  $n \geq 1$ , and any pair of rate- $R$  encoders  $f_1 : \mathbb{R}^{n \times a} \rightarrow [2^{naR}]$ ,  $f_2 : \mathbb{R}^{n \times b} \rightarrow [2^{nbR}]$  and decoder  $g : [2^{naR}] \times [2^{nbR}] \rightarrow \mathbb{R}^{a \times b}$ , we have

$$\mathbb{E} \|A^\top B - g(f_1(A), f_2(B))\|_F^2 \geq \sigma^4 \cdot nab \cdot \Gamma(R). \quad (7)$$

- 2) For any  $n \geq 1$ , and any rate- $R$  encoder  $f : \mathbb{R}^{n \times a} \rightarrow [2^{naR}]$  and decoder  $g : [2^{naR}] \times \mathbb{R}^{n \times b} \rightarrow \mathbb{R}^{a \times b}$ , we have

$$\mathbb{E} \|A^\top B - g(f(A), B)\|_F^2 \geq \sigma^4 \cdot nab \cdot 2^{-2R}. \quad (8)$$

In other words, the encoders  $f_1, f_2, g$  from Theorem 1 attain the lower bound from Theorem 2, and are therefore asymptotically optimal for this class of matrices.

We also show a simpler to use bound, based on a compression scheme that does not use a step of ‘‘MMSE scaling’’ and also does not use time-sharing. The resulting bound does not meet the lower bound of Theorem 2 for Gaussian iid matrices. However, for moderate  $R$  it is never much worse than the bound from Theorem 1. For some matrices  $A, B$  it is significantly better than the bound from Theorem 1.

*Theorem 3:* For any  $\varepsilon > 0$  and sufficiently large  $n$

- 1) There exist randomized encoders  $f_1 : \mathbb{R}^{n \times a} \rightarrow [2^{naR}]$ ,  $f_2 : \mathbb{R}^{n \times b} \rightarrow [2^{nbR}]$ , and decoder  $g : [2^{naR}] \times [2^{nbR}] \rightarrow \mathbb{R}^{a \times b}$ , such that for any  $A \in \mathbb{R}^{n \times a}$  and  $B \in \mathbb{R}^{n \times b}$  with bounded entries we have

$$\mathbb{E} \|A^\top B - g(f_1(A), f_2(B))\|_F^2 < \frac{\|A\|_F^2 \|B\|_F^2}{n} \left( \frac{2 \cdot 2^{2R} - 1}{(2^{2R} - 1)^2} + \varepsilon \right), \quad (9)$$

Furthermore, we have that

$$\mathbb{E}(C_{i,j} - \hat{C}_{i,j})^2 \leq \frac{\|a_i\|^2 \|b_j\|^2}{n} \left( \frac{2 \cdot 2^{2R} - 1}{(2^{2R} - 1)^2} + \varepsilon \right), \quad \forall i \in [a], j \in [b], \quad (10)$$

where  $C, \hat{C}, a_i, b_j$  are as in Theorem 1.

<sup>1</sup>corresponding to the case of ‘‘weights and attention’’ quantization and ‘‘weights-only’’ quantization in LLMs.

- 2) There exist a randomized encoder  $f : \mathbb{R}^{n \times a} \rightarrow [2^{naR}]$  and decoder  $g : [2^{naR}] \times \mathbb{R}^{n \times b} \rightarrow \mathbb{R}^{a \times b}$ , such that for any  $A \in \mathbb{R}^{n \times a}$  and  $B \in \mathbb{R}^{n \times b}$  with bounded entries we have

$$\mathbb{E} \|A^\top B - g(f(A), B)\|_F^2 < \frac{\|A\|_F^2 \|B\|_F^2}{n} \left( \frac{1}{2^{2R} - 1} + \varepsilon \right), \quad (11)$$

Furthermore,

$$\mathbb{E}(C_{i,j} - \hat{C}_{i,j})^2 \leq \frac{\|a_i\|^2 \|b_j\|^2}{n} \left( \frac{1}{2^{2R} - 1} + \varepsilon \right), \quad \forall i \in [a], j \in [b]. \quad (12)$$

where  $\hat{C} = g(f(A), B)$ .

Note that the term  $\|A^\top B\|_F^2$  does not appear at all in Theorem 3, and whenever  $\|A^\top B\|^2 \gg \frac{\|A\|_F^2 \|B\|_F^2}{n}$  the error in Theorem 3 is significantly smaller than the error in Theorem 1.

To put Theorem 3 in context, we note that recent work in LLMs suggested to use random rotation of  $A$  and  $B$  and then quantize each column of the rotated matrices using sub-optimal lattice quantizers [5], [6]. See more in Section I-C. In particular, a popular choice is to use the scalar quantizer, which is equivalent to quantizing to the lattice  $\mathbb{Z}^n$  with a cubic shaping region. In practice, to apply the scalar quantizer on a vector  $a_i \in \mathbb{R}^n$ , the common approach in the DNN and LLM literature is to store  $\|a_i\|_\infty$ , then normalize to  $\tilde{a}_i = a_i / \|a_i\|_\infty$  such that all entries of  $\tilde{a}_i$  are in  $[-1, 1]$ , then use a  $R$ -bit scalar quantizer with dynamic range  $[-1, 1]$ , and finally rescale the result by  $\|a_i\|_\infty$ . See, e.g. [6]. If the vector  $a_i$  is uniform on  $\sqrt{n}\mathbb{S}^{n-1}$  (the sphere with radius  $\sqrt{n}$ ), then for large  $n$  we have that  $\|a_i\|_\infty$  concentrates around  $\sqrt{2 \ln n}$ . It follows that the expected squared quantization error this quantizer attains per entry is  $(\frac{2}{3} \ln n) 2^{-2R}$ . Using this quantizer for matrix multiplication (after rotating each matrix by the same random rotation) therefore results in

$$\mathbb{E}(C_{i,j} - \hat{C}_{i,j})^2 \leq \frac{\|a_i\|^2 \|b_j\|^2}{n} \left( \frac{2}{3} \ln n \right) \left( \frac{2 \cdot 2^{2R} + \frac{2}{3} \ln n}{(2^{2R})^2} \right), \quad \forall i \in [a], j \in [b]. \quad (13)$$

Thus, replacing the scalar quantizer  $\mathbb{Z}^n$  with a high-dimensional pair of “good” nested lattices, as we do in the proof of Theorem 3 saves a factor of  $\frac{2}{3} \ln n$  in the expected squared error for moderate  $R$ .

The scheme used for proving Theorem 3 is based on using high-dimensional nested lattices with some asymptotically optimal properties. Unfortunately, such lattices do not lend themselves to efficient implementation. Another key contribution of this paper, described in Section VI, is a simplified nested-lattice quantization scheme, that is similar to the one used in the proof of Theorem 3, but is based on low-dimensional nested lattices. For such lattices, we provide a very fast implementation, whose efficiency does not depend on  $R$ . The simplified scheme attains performance fairly close to theoretical estimates therein. We believe the resulting quantization-based matrix multiplication scheme to be a good candidate for practical application in LLMs and other algorithms relying on heavy matrix multiplication operations and tolerant to approximation errors.

Additional contributions of this work include the following:

- We study the inner product case  $a = b = 1$ , in full generality, assuming the entries of  $A$  are drawn iid from distribution  $P$ , the entries of  $B$  are drawn iid from distribution  $Q$ , and the rates  $R_1$  and  $R_2$  are not necessarily equal. We derive several upper and lower bounds on the smallest attainable distortion in computing the inner product, and prove some results on the structure of the optimal encoders and decoder.
- For the matrix multiplication case, when the entries of  $A$  and  $B$  are drawn iid from a distribution  $P$  with zero mean and variance  $\sigma^2$ , we show that (7) continues to hold with  $\Gamma(R)$  replaced by  $\Gamma(R + D(P\|\mathcal{N}(0, \sigma^2)))$ .

The proofs of the main results are briefly sketched in Section I-B. We proceed to motivation and review of the literature.

### A. Importance of quantization for modern applications

To set the stage for the problem, let us estimate what level of quantization (in bits / entry) would be relevant for today’s main consumer of matrix multiplications: the large language models (LLMs). For those, quantization is typically employed for accelerating inference. During inference LLM is busy computing many products  $A^\top B$  of matrices with sizes  $d \times a$  and  $d \times b$  respectively. This requires  $2abd$  FLOPs and  $ad + bd + ab$  entries to load/store from memory. Ideally, we would want to quantize entries in such a way that all compute is fully utilized. For that we need to know the ratio  $\xi$  of available FLOPs to available memory bandwidth, a quantity known as “ops:bytes” of a processor. It ranges from  $\xi = 5 \dots 20$  for modern CPUs (FP32 arithmetic via AVX512) to  $\xi \approx 300$  for the

fastest GPUs (FP16 on an H100). The quantization rate saturating compute should then be bounded (in bits/entry) as

$$R < \frac{16}{\xi} \frac{ab}{a + b + \frac{ab}{d}}. \quad (14)$$

It turns out that there are two stages of running inference with LLMs: the pre-fill (when the input prompt is processed) and the generation (when response tokens are sequentially generated). During the pre-fill LLM we have  $a = d$  and  $b = L$  ( $d$  is the so-called hidden dimension and  $L$  is the sequence length), while during the generation we have  $a = L$  and  $b = 1$  (the  $A$  matrix coming from KV-cache and  $B$  matrix being new token's embedding). Thus, to saturate the computation core, we need

$$R_{\text{pre-fill}} = \frac{16Ld}{\xi(d + 2L)}, \quad R_{\text{generate}} = \frac{16L}{\xi(L + 1 + L/d)} \approx \frac{16}{\xi}.$$

We can see that during generation phase, on CPUs we would want to approach 1-3 bits/entry, while on GPUs we will not be able to ever saturate compute (that is, a decrease in quantization rate translates proportionally to decrease in runtime). For the pre-fill phase, for large LLMs we get  $R_{\text{generate}} > 16$  bit (that is, just storing plain FP16 is already good enough). Quantization is still important for “small” LLMs running on fast (H100-like) GPUS: for example, for BERT [7] we have  $L = 512$ ,  $d = 768$  and  $\xi = 300$  (for an H100), resulting in pre-fill phase requiring quantization rate  $R \approx 11.7$  bit/entry.

### B. Sketch of the proof

This work started with the goal of trying to understand approximate matrix multiplication for two matrices  $A$  and  $B$  which are random, with iid Gaussian entries  $\mathcal{N}(0, 1)$ . We started by trying to solve the case of  $a = b = 1$  (Sections II and III), i.e. when  $A^\top B$  is simply an inner product of two iid Gaussian vectors.

Recall that the Gaussian distortion-rate function  $D(R) = 2^{-2R}$ , e.g. [8, Section 26.1.2]. A simple argument (Thm. 5) shows that compressing  $A$  to  $\hat{A}$  and  $B$  to  $\hat{B}$  via rate- $R$  optimal Gaussian vector quantizer achieves error

$$\mathbb{E}[(\hat{A}^\top \hat{B} - A^\top B)^2] \leq \phi(D(R)), \quad \phi(x) := 2x - x^2.$$

It turned out that the function  $\phi(D(R))$  is monotonically decreasing but *not* convex. Thus, via time-sharing one can achieve a lower convex envelope of  $\phi(D(R))$ , which turns out to be the  $\Gamma(R)$  function defined in (1).

We next proceed to lower bounds on distortion or, equivalently, to upper bounds on rate  $R$  required for the existence of encoders  $f_1, f_2$  and decoder  $g$  satisfying

$$\mathbb{E}[(g(f_1(A), f_2(B)) - A^\top B)^2] \leq nD \quad (15)$$

A simple oracle bound (by revealing  $B$  to the decoder) shows that rate  $R$  cannot be smaller than the standard Shannon rate-distortion function of  $A$ , see Theorem 4. However, this bound leaves a wide gap with the achievability bound given above. Next, by a standard data-processing argument (and observation that encoders for  $A$  and  $B$  can be without loss of generality be taken identical) in Section III-B we deduce that 15 requires rate

$$R \geq \limsup_{n \rightarrow \infty} \frac{1}{n} \inf_{\hat{A}} \{I(A; \hat{A}) : \frac{1}{n} \sum_{i=1}^n \phi(\lambda_i) \leq D\}, \quad (16)$$

where  $A \sim \mathcal{N}(0, I_n)$ , infimum is over all  $\mathbb{R}^n$ -valued random variables  $\hat{A}$  and  $\{\lambda_i\}$  are the eigenvalues of  $\text{Cov}(A|\hat{A})$ . This reduces inner-product quantization to an optimization of a multi-letter mutual information. Notice that the distortion constraint is no longer separable, and hence the standard single-letterization (e.g. [8, Theorem 24.8]) does not work and the limit on the right-hand side is not possible to evaluate. For the special case of Gaussian distribution of entries of  $A$  we were able to single-letterize the expression on the right-hand side of (16), see Theorem 6, showing that left-hand side of (16) evaluates to  $\Gamma^{-1}(D)$ . Putting both upper and lower bounds together, we conclude that optimal compression rate for the iid Gaussian inner-product problem is thus given by  $\Gamma^{-1}(D)$ , see Theorem 7.

We next proceed to solving the matrix case. Luckily, it turns out that for Gaussian iid matrices, again, the optimal compression for matrix multiplication of  $A^\top B$  is asymptotically achieved by compressing each column separately via the use of optimal inner-product quantizers, see Theorems 8 and 9.

Having solved the iid Gaussian case, we “derandomize” our constructions by following the work of [9]. Specifically, for the inner product problem, applying the same random orthogonal rotation to two arbitrary vectors we can

create vectors whose marginal distribution is uniform on the sphere, while their inner product is unchanged by the random rotation. The important observation is that a high-dimensional vector that is uniform on the sphere is very similar to an iid Gaussian vector (for example, in terms of joint distribution of small  $O(\sqrt{n})$ -sized subsets). Thus, after random rotation we may hope to achieve compression performance of iid Gaussian matrices. Indeed, using a “good” nested lattice quantizer, in the sense of [9], we can obtain a reconstruction error that depends only on the inner product between the vectors and their individual  $\ell_2$  norms, see Theorems 11 and 12. Since the performance bounds coincides with the lower bound for the iid Gaussian case, it turns out that the resulting quantizers are optimal and generally cannot be improved (except in terms of finite- $n$  dependence).

Together these steps complete proof of the main results quoted above.

### C. Related work

Randomized linear algebra/sketching, and locality-sensitive hashing (LSH) are techniques widely used in practice, to compute approximate inner products and approximate matrix multiplications, as well as other operations, in reduced dimensions. The figure of merit in these fields is typically the tradeoff between the reduced dimension and the approximation error. Since the dimension of the reduced matrix/vector is related to the number of bits required for storing it, this body of work is relevant to our study. However, the tradeoff between the number of bits per dimension and the total approximation error, and its dependence on the properties of  $A$ ,  $B$  and  $A^\top B$  is often subtle. Thus, there is no immediate translation between the required dimension of a sketch and the number of bits needed for representing it for obtaining the required accuracy.

Many algorithms have been developed for randomized linear algebra, see [10], [11] for a survey, and in particular for approximate matrix multiplication. A canonical example is the Monte-Carlo Matrix Multiplication (MCMM) algorithm [12] which randomly samples (the same)  $c$  rows from  $A \in \mathbb{R}^{n \times a}$  and  $B \in \mathbb{R}^{n \times b}$  and estimates  $A^\top B$  as the (scaled) matrix multiplication of the sub-sampled matrices. Thus, each matrix is represented by  $ac$  (respectively  $bc$  real numbers), and the expected squared Frobenius norm of the approximation error is shown to scale like  $O(\|A\|_F^2 \|B\|_F^2 / c)$ . Similarly, LSH for cosine similarity or  $\ell_2$  distance also produce low-dimensional sketches of  $u \in \mathbb{R}^n$  and  $v \in \mathbb{R}^n$ , from which the inner product of  $u^\top v$  can be approximated. Specifically, in [13] it is proposed to project the two vectors using  $c$  random projections (same random projections for both vectors) and only store the sign of each projection. The Hamming distance between the obtained vectors is distributed as Binomial  $\left(c, \frac{\theta(u,v)}{\pi}\right)$  where<sup>2</sup>  $\theta(u,v) = \cos\left(\frac{u^\top v}{\|u\| \|v\|}\right)$ , such that the expected squared error in estimating  $\theta(u,v)$  is  $O(1/c)$ . In [14] it is proposed to estimate  $\|u - v\|$  (which is equivalent to estimating  $u^\top v$  for  $u$  and  $v$  on the sphere) by computing Gaussian linear combinations of each of them, and using a (dithered) scalar quantizer for quantizing each of the entries of the linear combinations. Specifically, for a vector  $G \in \mathbb{R}^n$ , with iid  $\mathcal{N}(0,1)$  entries, we have that  $G^\top(u - v) \sim \mathcal{N}(0, \|u - v\|^2)$ , and therefore the probability that  $G^\top u$  and  $G^\top v$  are quantized (after dithering) to the same value is a monotone function of  $\|u - v\|$ .

All the schemes mentioned above, as well as many other sketching/LSH schemes suffer from the same shortcoming: their relative error  $\frac{\mathbb{E}\|\widehat{A^\top B} - A^\top B\|_F^2}{\|A^\top B\|_F^2}$  scales like  $O\left(\frac{1}{c} \frac{\|A\|_F^2 \|B\|_F^2}{\|A^\top B\|_F^2}\right)$ , and typically these schemes are applied with constant  $c$ . When  $\frac{\|A\|_F^2 \|B\|_F^2}{\|A^\top B\|_F^2} = \Omega(1)$ , these schemes perform remarkably well, despite the fact that  $c$  does not grow with  $n$ . However, when  $\frac{\|A\|_F^2 \|B\|_F^2}{\|A^\top B\|_F^2} = \omega(1)$ , as is the case for random iid matrices, their relative error is very high. A notable exception is the algorithm proposed by Pagh in [15], which represents each matrix using  $n \cdot \min\{m, a\}$  (respectively  $n \cdot \min\{m, b\}$ ) real numbers, and produces an *unbiased* estimator for  $A^\top B$  with expected error of  $\mathbb{E}\left(\widehat{(A^\top B)}_{i,j} - (A^\top B)_{i,j}\right)^2 = O\left(\frac{\|A^\top B\|_F^2}{m}\right)$ , for all  $i, j$ , and does so with runtime proportional to  $n^2 + nm$  (ignoring logarithmic factors). When the product  $A^\top B$  is known to be highly sparse, this allows to estimate the sparsity pattern with  $m$  proportional to the number of nonzero entries.

The topic of matrix quantization has received much attention in the last decade in the context of DNNs and LLMs. The goal here is to reduce the memory footprint of the weight matrices, allowing to load them to the main memory using less IOs, as well as speed up the multiplications and additions operations by moving from floating point numbers to small integers (and when possible, also sparsifying the matrices, saving some operations altogether). Roughly speaking, one can distinguish between two paradigms: *quantization-aware training*, where the training procedure is designed to output weight matrices with “cheap” representation [16], [17], and *post-training*

<sup>2</sup>In this paper, whenever the type of the norm  $\|\cdot\|$  is not specified, it is the Euclidean ( $\ell_2$ ) norm.

*quantization*, where the training procedure is performed in high precision, and quantization of the weights is only performed after training has terminated (perhaps with some fine tuning afterwards) [18], [19], [20], [21], [22], [23], [5]. In order to further speed up matrix multiplication, some works also develop quantizers for the activations [19], [21], [22], [23], while other works assume the activations are kept in high precision [18], [5]. Quantization for DNNs and LLMs are typically evaluated according to the end-to-end performance of the quantized architecture, but often the Frobenius norm of the approximation error is considered as the intermediate optimization criterion for quantizing the weights at each layer [16], [24]. Some works rely on specific empirical observations on the distribution of weights and activations in LLMs. For example [20], [21], [22] exploit the fact that a small subset of entries in the activations have significantly larger magnitude than the majority of entries. Most notably, in [23] it is observed that for large LLMs, quantizing all weights to  $\{-1, 0, 1\}$  and the activations to 8 bits, hardly affects the overall performance. Among the work described above, the algorithm from [5] is closest to the scheme we use in the proof of our Theorem 1 and Theorem 3, as well as the practical adaptation of the scheme used in those proofs, which is described in Section VI. The work [5] develops an algorithm for quantizing the weight matrices (keeping the activations in high precision) that is based on random rotation using the randomized Hadamard transform (that can be performed in time  $n \log n$ ) and then using the  $E_8$  lattice for quantizing the rotated matrix. The mapping from lattice points to bits that was used in [5] required access to a lookup table, and was tailor-designed for  $R = 2$ , while using different rates requires to further use successive refinement. While our practical scheme in Section VI also uses product-lattice quantization, we use a nested lattice quantizer, which results in a simple mapping from lattice points to bits, regardless of  $R$ . Furthermore, we quantize both matrices to be multiplied. In addition to reducing the limitations incurred by the memory bandwidth, an additional benefit of quantizing both matrices, is that one can replace the decoder with a lookup table, as in [25], [26], [27], [28], resulting in very fast decoding in CPUs. Furthermore, our quantizers are *universal* - that is, they do not depend on the distribution of the entries in the matrices to be multiplied, and the approximation error depends only on the Frobenius norms of these matrices. The quantizer design procedure in [5], on the other hand, utilizes the empirical distribution of the activations to some extent. Following [5], the QuaRot [6] scheme also uses randomized Hadamard transform prior to quantization, followed by 4-bit scalar quantization of each entry in both rotated matrices. Our implementation in Section VI quantizes the entries of the rotated matrices using nested-lattice codes, which come much closer to the optimal rate-distortion tradeoff than scalar quantizers, with essentially the same complexity (provided that the base lattice has an efficient nearest-neighbor decoder, as is the case for essentially all “good” lattices in dimensions 2, 3, 4 and 8).

To the best of our knowledge, there was very little work on distributed compression for inner product/matrix multiplication in the information theory literature. Recently, Malak [29] studied the problem of *lossless* distributed compression of binary random matrices for computing their product, and derived non-trivial bounds under stringent assumptions on the joint distribution. Some prior work considered the problem of distributed compression of random vectors with the goal of approximately computing a linear function of those vectors [30], [31]. In those works, the goal was to estimate, say, the difference between the two vectors in  $\mathbb{R}^n$ , which is itself a vector in  $\mathbb{R}^n$ . While the inner product of these vectors, which is a scalar in  $\mathbb{R}$ , can be computed from their difference (assuming their individual norms were encoded in high resolution), it seems, a-priori, that distributed compression for inner-product computation is an easier task. Our results show that this is, in fact, hardly the case. Another line of related work in the information theory literature, is that of Ingber et al. [32] that considered the fundamental limits of lossy compression of a database in order to support approximate nearest neighbor search (see also [33] for a practical implementation). We note in passing that much recent work focused on coding for speeding up distributed matrix multiplication by introducing redundancy for mitigating the effect of “slow workers” (stragglers), see, e.g., [34]. This line of work is not directly related to approximate matrix multiplication via compression, studied in this paper.

Finally, one may wonder if approximating matrix product in mean squared error (MSE) metric is the right distortion metric. Indeed, it was shown in [35] that if the high-dimensional vectors to be compressed are probability distributions and the metric is KL divergence (reasonable assumptions for attention matrices in LLMs), the optimal quantization algorithms become quite different from the MSE ones. We leave these extensions for future work.

To summarize, the main innovations of this work with respect to prior work are:

- a. Our work provides, for the first time, the fundamental limits of quantization for matrix multiplication. We derive a non-asymptotic lower bound on the error of any quantization algorithm for the case of Gaussian iid matrices, and develop a “robust” quantization algorithm (that makes no assumptions on the matrices  $A, B$ ) that asymptotically attains it. This gives a useful benchmark for evaluating the performance of any robust quantization algorithm.

- b. Our lower and upper bounds give a theoretic justification for the widely used idea of applying the same random rotation to both matrices  $A$  and  $B$  prior to quantization. In particular, the schemes used in the proofs of Theorem 1 and Theorem 3 are based on random rotation followed by quantizers based on “good” high-dimensional nested lattices. The scheme from the Proof of Theorem 3 improves upon the QuaRot scheme [6] by employing “good” lattices instead of  $\mathbb{Z}^n$  (scalar quantizer), which leads to a multiplicative reduction by of factor  $O(\log n)$  in the resulting distortion, see (13).
- c. Motivated by the large gap between the performance of the scheme from Theorem 3 and that of a similar scheme that uses scalar quantizers, in Section VI we describe a family of nested lattice quantization schemes that on the one hand admit very fast implementation, and on the other hand attain performance that comes close to the that of Theorem 3 (which is nearly optimal for Gaussian iid matrices). Our scheme is based on using a lattice  $\Lambda$  in relatively small dimension for both quantization and “shaping”.<sup>3</sup> We describe quantizers with a very fast implementation for any  $R > 0$ , which only require access to a fast nearest neighbor decoder for the lattice  $\Lambda$  (for many important lattices, e.g., the  $D_n$  and  $A_n$  family,  $E_8$ , etc., fast nearest neighbor decoders are known). While for weights-only quantization the work [5] already proposed using random rotation followed by quantization using the  $E_8$  lattice, the shaping mechanism used in [5] is based on using a lookup table, which limits the  $R$  it can work with,<sup>4</sup> and is also less efficient on a GPU. Our proposed shaping mechanism is much faster, regardless of the quantization rate  $R$ . Furthermore, we propose in Section VI a mechanism for controlling overload, which plays an important role in coming close to the performance of optimal high-dimensional nested lattices using simple low-dimensional lattices instead.

#### D. Paper organization

We begin our study with the special case where  $a = b = 1$ , so that matrix multiplication becomes an inner product. The reason is twofold: First, it is easier to gain intuition for this problem, and all techniques for proving converse (impossibility) results for the inner product case, easily extend to the matrix multiplication case. The second reason is that our achievability results are based on compression of each column of  $A$  separately and compression of each column of  $B$  separately, and estimating each inner product forming  $C_{ij} = (A^\top B)_{i,j} = a_i^\top b_j$  separately. In Section II we formally define the compression for inner product computation problem, identify the structure of the optimal decoder, and give simple expressions on the attained distortion as a function of the encoders  $f_1$  and  $f_2$ , as well as a simple lower bound on the distortion in terms of the “standard” distortion-rate function. In Section III we restrict attention to the symmetric case where the two vectors have the same distribution, and both encoders have the same rate. We prove lower and upper bounds on the smallest attainable distortion in this case, which coincide in the Gaussian case. In Section IV we generalize the inner-product results for matrix multiplication  $A^\top B$  of  $A \in \mathbb{R}^{n \times a}$  and  $B \in \mathbb{R}^{n \times b}$ , for general  $a$  and  $b$ . Building on the bounds developed for the inner-product case of  $a = b = 1$ , we prove lower and upper bound on the smallest expected squared Frobenius norm of the error. In the special case where all entries in both matrices are iid Gaussian, the lower bound results in Theorem 2. In Section V we develop a quantization scheme, based on randomly rotating both  $A$  and  $B$  by the same rotation matrix, and then using nested-lattice quantizers for separately quantizing each column of the rotated matrices, for quantization of arbitrary matrices  $A \in \mathbb{R}^{n \times a}$  and  $B \in \mathbb{R}^{n \times b}$ . The expected squared Frobenius norm of the approximation error attained by this scheme is upper bounded in Theorem 1. The upper bound depends on  $A$  and  $B$  only through  $\|A\|_F$ ,  $\|B\|_F$ ,  $\|A^\top B\|_F$ , and meets the lower bound from Theorem 2 for the case where  $A$  and  $B$  have Gaussian iid entries. The estimation of  $\widehat{A^\top B}$  formed by the scheme used in the proof of Theorem 1 is biased. We also introduce in Section V a very simple variation of the same scheme, that results in an unbiased estimator for  $\widehat{A^\top B}$ , whose expected squared Frobenius norm error is upper bounded in Theorem 3, in terms of only  $\|A\|_F$ ,  $\|B\|_F$ . Finally, in Section VI we introduce a practical implementation of the quantization scheme from Theorem 3 for matrix multiplication of arbitrary matrices. In these scheme, we describe several compromises in the choice of lattices used for quantization, as well as in the rotation matrix used for rotating both  $A$  and  $B$ . With these compromises the quantization scheme and the decoder become extremely simple and fast. Some numerical evidence show that, nevertheless, the resulting approximation error is quite close to the lower bound from Theorem 2.

<sup>3</sup>A lattice has an infinite number of points. In order to use lattice quantizers as a mean of converting vectors in  $\mathbb{R}^d$  to  $dR$  bits, one must choose a set of  $2^{dR}$  points of the lattice which are the codewords in the quantization codebook. The operation of selecting those points is called shaping.

<sup>4</sup>For large  $R$ , [5] proposes to reuse the codebook for small  $R$  via a successive refinement procedure. This adds further delays to the quantization process.



### E. Notation

For  $x > 0$  we denote by  $\log(x)$  the logarithm of  $x$  taken to base 2, and by  $\ln(x)$  the natural logarithm. We denote the Euclidean ( $\ell_2$ ) norm of a vector  $x \in \mathbb{R}^n$  by  $\|x\| = \sqrt{\sum_{i=1}^n x_i^2}$  and its  $\ell_1$  norm by  $\|x\|_1 = \sum_{i=1}^n |x_i|$ . For a matrix  $A \in \mathbb{R}^{n \times m}$  the Frobenius norm is  $\|A\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m A_{ij}^2} = \sqrt{\text{trace}(A^\top A)}$ . The multiset of eigenvalues of a square matrix  $A \in \mathbb{R}^{n \times n}$  is denote by  $\text{eig}(A) = (\lambda_1, \dots, \lambda_n)$ . For  $y > 1$  we denote  $\lfloor y \rfloor = \{1, \dots, \lfloor y \rfloor\}$ .

## II. COMPRESSION FOR INNER-PRODUCT COMPUTATION: GENERAL PROBLEM SETUP AND SIMPLE BOUNDS

Let  $P$  and  $Q$  be distributions on  $\mathbb{R}$  with unit variance, and let  $U \sim P^{\otimes n}$  and  $V \sim Q^{\otimes n}$  be statistically independent. We consider the problem of quantizing  $U$  and  $V$  in order to compute their inner product  $U^\top V$ . In particular, an  $(n, R_1, R_2, D)$  code consists of mappings

$$f_1 : \mathbb{R}^n \rightarrow [2^{nR_1}] \quad (17)$$

$$f_2 : \mathbb{R}^n \rightarrow [2^{nR_2}] \quad (18)$$

$$g : [2^{nR_1}] \times [2^{nR_2}] \rightarrow \mathbb{R}, \quad (19)$$

with

$$D = D^{\text{IP}} = \frac{1}{n} \mathbb{E} (U^\top V - g(f_1(U), f_2(V)))^2. \quad (20)$$

We define

$$D_n^{\text{IP},*}(R_1, R_2) = D_n^{\text{IP},*}(R_1, R_2, P, Q) = \inf \{D : \exists(n, R_1, R_2, D) \text{ - code}\}. \quad (21)$$

We further define the asymptotic function

$$D^{\text{IP}}(R_1, R_2) = D^{\text{IP}}(R_1, R_2, P, Q) = \limsup_{n \rightarrow \infty} D_n^*(R_1, R_2). \quad (22)$$

Note that the assumption that  $P$  and  $Q$  have unit variance is without loss of generality. To see this, assume that  $\tilde{U} \sim \tilde{P}^{\otimes n}$  and  $\tilde{V} \sim \tilde{Q}^{\otimes n}$ , such that  $\text{Var}[\tilde{U}] = \sigma_1^2$  and  $\text{Var}[\tilde{V}] = \sigma_2^2$ , and we would like to quantize  $\tilde{U}$  and  $\tilde{V}$  in order to estimate  $\tilde{U}^\top \tilde{V}$ . To that end we may define the unit-variance random variables  $U = \frac{\tilde{U}}{\sigma_1}$  and  $V = \frac{\tilde{V}}{\sigma_2}$  with corresponding distributions  $P$  and  $Q$ , compress them using  $f_1(U)$  and  $f_2(V)$ , and estimate the inner product as

$$\widehat{\tilde{U}^\top \tilde{V}} = \sigma_1 \sigma_2 \cdot g(f_1(U), f_2(V)), \quad (23)$$

where  $f_1, f_2, g$  attain  $D_n^{\text{IP},*}(R_1, R_2)$  for  $P$  and  $Q$ . This scheme will achieve

$$\mathbb{E} \left( \tilde{U}^\top \tilde{V} - \widehat{\tilde{U}^\top \tilde{V}} \right)^2 = \sigma_1^2 \sigma_2^2 \cdot \mathbb{E} (U^\top V - g(f_1(U), f_2(V)))^2 = \sigma_1^2 \sigma_2^2 \cdot D_n^{\text{IP},*}(R_1, R_2). \quad (24)$$

This must be optimal, since otherwise we could have attained a smaller distortion for  $P$  and  $Q$  by first scaling  $U$  and  $V$  by  $\sigma_1$  and  $\sigma_2$ , respectively, feeding them to the better inner product quantization system, and scaling the obtained inner product estimate by  $\frac{1}{\sigma_1 \sigma_2}$ .

Some of our bounds will rely on the distortion-rate function of a source on  $\mathbb{R}$  under quadratic distortion. An  $(n, R, D)$  code for a source  $U \sim P^{\otimes n}$  consists of an encoder  $f : \mathbb{R}^n \rightarrow [2^{nR}]$  and a decoder  $g : [2^{nR}] \rightarrow \mathbb{R}^n$  with  $D = \frac{1}{n} \mathbb{E} \|U - g(f(U))\|^2$ . We denote by  $D_n^*(R) = D_n^*(R, P)$  the smallest distortion attained by any  $(n, R, D)$  code, and we denote the distortion-rate function by [8]

$$D_P(R) = \lim_{n \rightarrow \infty} D_n^*(R, P) = \min_{P_{Y,U}: I(U;Y) \leq R} \mathbb{E}(U - Y)^2. \quad (25)$$

It is also well-know [8], that  $D_n^*(R, P) \geq D_P(R)$  for any  $n \geq 1$ .

### A. Optimal Decoder and Error Expressions

In the following, we assume  $f_1$  and  $f_2$  are fixed. We denote  $W_U = f_1(U)$  and  $W_V = f_2(V)$ . Let  $\hat{U} = \mathbb{E}[U|W_U]$  and  $\hat{V} = \mathbb{E}[V|W_V]$ .

*Proposition 1:* The optimal choice for  $g$  is  $g^*(W_U, W_V) = \hat{U}^\top \hat{V}$ .

**Proof.** The minimum mean squared estimator (MMSE) of a random variable  $X$  from another random variable  $Y$  is  $\hat{X} = \mathbb{E}[X|Y]$ . Thus,

$$g^*(W_U, W_V) = \mathbb{E}[U^\top V | W_U, W_V] = \mathbb{E}[U^\top | W_U] \mathbb{E}[V | W_V] = \hat{U}^\top \hat{V}, \quad (26)$$

where the second equality follows since  $(U, W_U) \perp\!\!\!\perp (V, W_V)$ . ■

Let  $e_U = U - \hat{U}$  and  $\Sigma_{e_U} = \mathbb{E}[(U - \hat{U})(U - \hat{U})^\top]$ . Similarly, let  $e_V = V - \hat{V}$  and  $\Sigma_{e_V} = \mathbb{E}[(V - \hat{V})(V - \hat{V})^\top]$ . Recall that by the orthogonality principle [36, Chapter 4.2], it holds that  $\mathbb{E}[\hat{U} e_U^\top] = 0$  and  $\mathbb{E}[\hat{V} e_V^\top] = 0$ .

*Proposition 2:* For the optimal decoder, we have that

$$\mathbb{E} (U^\top V - g^*(W_U, W_V))^2 = \frac{1}{n} [\text{trace}(\Sigma_{e_U}) + \text{trace}(\Sigma_{e_V}) - \text{trace}(\Sigma_{e_U} \Sigma_{e_V})] \quad (27)$$

**Proof.** We have

$$D^{\text{IP}} = \mathbb{E} \left( (\hat{U} + e_U)^\top (\hat{V} + e_V) - \hat{U}^\top \hat{V} \right)^2 \quad (28)$$

$$= \mathbb{E} \left( \hat{U}^\top e_V + \hat{V}^\top e_U + e_U^\top e_V \right)^2 \quad (29)$$

$$= \mathbb{E} \left( \hat{U}^\top e_V \right)^2 + \mathbb{E} \left( \hat{V}^\top e_U \right)^2 + \mathbb{E} \left( e_U^\top e_V \right)^2, \quad (30)$$

where the last transition is due to the orthogonality principle and the statistical independence of  $U$  and  $V$ . We have that

$$\mathbb{E} \left( \hat{U}^\top e_V \right)^2 = \text{trace} \left[ \mathbb{E}[\hat{U} \hat{U}^\top e_V e_V^\top] \right] = \text{trace} \left[ \mathbb{E}[\hat{U} \hat{U}^\top] \Sigma_{e_V} \right] \quad (31)$$

Recalling that  $\mathbb{E}[\hat{U} \hat{U}^\top] = I - \Sigma_{e_U}$ , again, by the orthogonality principle, we obtain

$$\mathbb{E} \left( \hat{U}^\top e_V \right)^2 = \text{trace} [(I - \Sigma_{e_U}) \Sigma_{e_V}] = \text{trace}(\Sigma_{e_V}) - \text{trace}(\Sigma_{e_U} \Sigma_{e_V}), \quad (32)$$

Similarly,

$$\mathbb{E} \left( \hat{V}^\top e_U \right)^2 = \text{trace}(\Sigma_{e_U}) - \text{trace}(\Sigma_{e_U} \Sigma_{e_V}). \quad (33)$$

Finally,

$$\mathbb{E} \left( e_U^\top e_V \right)^2 = \text{trace} \left[ \mathbb{E}[e_U e_U^\top e_V e_V^\top] \right] = \text{trace}(\Sigma_{e_U} \Sigma_{e_V}). \quad (34)$$

■

### B. Simple Lower Bounds

We show that computing the inner product with mean squared error (MSE) of  $nD$  is necessarily harder than compressing each of the random vectors  $U$  and  $V$  with  $\ell_2$  norm of  $nD$ . Note that in the inner product quantization problem we are only interested in a single scalar in  $\mathbb{R}$  while in the standard problem of quantizing a random vector we are interested in a vector in  $\mathbb{R}^n$ . Yet, the former problem is at least as hard as the latter.

*Theorem 4:* For any  $n \geq 1$

$$D_n^{\text{IP},*}(R_1, R_2, P, Q) \geq \max \{D_P(R_1), D_Q(R_2)\}, \quad (35)$$

and in particular

$$D^{\text{IP}}(R_1, R_2, P, Q) \geq \max \{D_P(R_1), D_Q(R_2)\}. \quad (36)$$

**Proof.** From Proposition 2 we have that for any  $f_1 : \mathbb{R}^n \rightarrow [2^{nR_1}]$  and  $f_2 : \mathbb{R}^n \rightarrow [2^{nR_2}]$

$$\mathbb{E} (U^\top V - g^*(W_U, W_V))^2 = \frac{1}{n} [\text{trace}(\Sigma_{e_V}) + \text{trace}(\Sigma_{e_U}) - \text{trace}(\Sigma_{e_U} \Sigma_{e_V})] \quad (37)$$

$$= \frac{1}{n} \left[ \text{trace}(\Sigma_{e_U}) + \mathbb{E}(\hat{U}^\top e_V)^2 \right] \quad (38)$$

$$\geq \frac{1}{n} \text{trace}(\Sigma_{e_U}) \quad (39)$$

$$\geq D_n^*(R_1, P), \quad (40)$$

where (38) follows from (32), and the last inequality follows since  $W_U$  is an encoding of  $U$  with  $2^{nR_1}$  codewords, which must incur distortion at least  $D_n^*(R_1, P)$  by definition. Note that the inequality (39) holds with equality in the “single-sided” case where only  $U$  is quantized while  $\hat{V} = V$ , so that  $e_V = 0$ . The bound  $D_n^*(R_1, R_2, P, Q) \geq D_n^*(R_2, Q)$  follows similarly. Our statement now follows since  $D_n^*(R_1, P) \geq D_P(R_1)$  and  $D_n^*(R_2, Q) \geq D_Q(R_2)$  for any  $n \geq 1$ . ■

### III. COMPRESSION FOR INNER-PRODUCT COMPUTATION: THE SYMMETRIC CASE

In this section we assume  $P = Q$ ,  $R_1 = R_2 = R$ , and define  $D_n^{\text{IP},*}(R, P) = D_n^*(R, R, P, P)$ , and  $D^{\text{IP},*}(R, P) = D^{\text{IP},*}(R, R, P, P)$ . We first develop a simple upper bound based on using the same encoder for both vectors (that is  $f = f_1 = f_2$ ), that time-shares between a “good” encoder for  $P$  under quadratic distortion, and a zero-rate encoder. We then develop a lower bound on the distortion of inner-product compression, which shows that for the symmetric case, using the same encoder  $f = f_1 = f_2$  for both  $U$  and  $V$  is optimal, and depends on the spectrum of the covariance matrix of  $e_U = U - \mathbb{E}[U|f(U)]$ . We then give some constraints on the error spectrum that can be attained by a rate  $R$  encoder. Using this characterization we obtain a general lower on  $D^{\text{IP},*}(R, P)$  which meets the upper bound when  $P$  is the Gaussian distribution.

#### A. Upper Bound

Define the function

$$\phi(x) = 2x - x^2. \quad (41)$$

and note that  $x \mapsto \phi(x)$  is increasing and concave on  $[0, 1]$ . We give a time-sharing upper bound on  $D^{\text{IP},*}(R, P)$  in terms of  $\phi(D_P(R))$ .

*Theorem 5:*

$$D^{\text{IP},*}(R, P) \leq \min_{0 \leq \alpha \leq 1} (1 - \alpha) + \alpha \cdot \phi \left( D_P \left( \frac{R}{\alpha} \right) \right) \quad (42)$$

**Proof.** We will prove that

$$D_n^{\text{IP},*}(R, P) \leq \min_{\alpha \in \frac{1}{n} \{0, 1, \dots, n\}} (1 - \alpha) + \alpha \cdot \phi \left( D_{\alpha n}^* \left( \frac{R}{\alpha}, P \right) \right) \quad (43)$$

from which the statement immediately follows. Let  $\alpha \in \frac{1}{n} \{0, 1, \dots, n\}$ , and consider a compressor for  $P^{\otimes \alpha n}$  under quadratic distortion:  $f : \mathbb{R}^{\alpha n} \rightarrow [2^{nR} = 2^{n\alpha \frac{R}{\alpha}}]$  and  $g : [2^{nR} = 2^{n\alpha \frac{R}{\alpha}}] \rightarrow \mathbb{R}^{\alpha n}$ , that attains

$$D = \frac{1}{\alpha n} \mathbb{E} \|U^{\alpha n} - \hat{U}^{\alpha n}\|^2 = \frac{1}{\alpha n} \text{trace}(\Sigma_{e_{U^{\alpha n}}}). \quad (44)$$

We encode  $U$  by applying  $f$  on  $U^{\alpha n}$  and do not describe the other coordinates. The resulting covariance error matrix is therefore block diagonal of the form

$$\Sigma_{e_U} = \begin{bmatrix} \Sigma_{e_{U^{\alpha n}}} & 0 \\ 0 & I_{(1-\alpha)n} \end{bmatrix}. \quad (45)$$

Consequently,

$$\text{trace}(\Sigma_{e_U}) = \text{trace}(\Sigma_{e_{U^{\alpha n}}}) + \text{trace}(I_{(1-\alpha)n}) = n\alpha D + n(1 - \alpha) \quad (46)$$

$$\text{trace}(\Sigma_{e_U} \Sigma_{e_U}) = \text{trace}(\Sigma_{e_{U^{\alpha n}}} \Sigma_{e_{U^{\alpha n}}}) + \text{trace}(I_{(1-\alpha)n}) = \|\Sigma_{e_{U^{\alpha n}}}\|_F^2 + n(1 - \alpha). \quad (47)$$

Recall that for a positive semi-definite matrix  $A \in \mathbb{R}^{m \times m}$  it holds that  $\|A\|_F^2 \geq \frac{1}{m}(\text{trace}(A))^2$ . This follows since the vector  $\lambda = \text{eig}(A)$  has non-negative entries, so that  $\text{trace}(A) = \|\lambda\|_1$ , and therefore  $\|A\|_F^2 = \|\lambda\|_2^2 \geq \frac{1}{m}\|\lambda\|_1^2 = \frac{1}{m}(\text{trace}(A))^2$ . Thus,

$$\|\Sigma_{e_U \alpha_n}\|_F^2 \geq \frac{1}{\alpha n} (\text{trace}(\Sigma_{e_U \alpha_n}))^2 = \alpha n D^2, \quad (48)$$

and, by (47), we have

$$\text{trace}(\Sigma_{e_U} \Sigma_{e_V}) \geq n \alpha D^2 + n(1 - \alpha). \quad (49)$$

We use the same encoder also for encoding  $V$ , such that  $\Sigma_{e_V} = \Sigma_{e_U}$ , and use the optimal decoder  $g^*$  for estimating  $U^\top V$ . Applying Proposition 2, we obtain

$$D^{\text{IP}} = \frac{1}{n} [\text{trace}(\Sigma_{e_U}) + \text{trace}(\Sigma_{e_V}) - \text{trace}(\Sigma_{e_U} \Sigma_{e_V})] \quad (50)$$

$$= \frac{1}{n} [2 \text{trace}(\Sigma_{e_U}) - \|\Sigma_{e_U}\|_F^2] \quad (51)$$

$$\leq (1 - \alpha) + \alpha \cdot (2D - D^2) \quad (52)$$

$$= (1 - \alpha) + \alpha \phi(D). \quad (53)$$

Taking the compressor  $f$  that attains  $D_{\alpha n}^*(\frac{R}{\alpha}, P)$ , we obtain the claimed result. ■

## B. Lower Bound

*Lemma 1:* For the symmetric case there is no loss of optimality in taking  $f_1 = f_2 = f$ , and

$$D_n^{\text{IP},*}(R, P) = \frac{1}{n} \inf_f [2\|\lambda(f)\|_1 - \|\lambda(f)\|_2^2] = \frac{1}{n} \inf_f \sum_{i=1}^n \phi(\lambda_i(f)), \quad (54)$$

where the infimum runs over all encoders  $f : \mathbb{R}^n \rightarrow [2^{nR}]$ , and

$$\lambda(f) = \text{eig}(\Sigma_{e_U^f}), \quad (55)$$

where  $e_U^f = U - \mathbb{E}[U|f(U)]$ ,  $\Sigma_{e_U^f} = \mathbb{E}[e_U^f e_U^{f\top}]$ .

**Proof.** By Proposition 2, we have that for any two encoders  $f_1 : \mathbb{R}^n \rightarrow [2^{nR}]$  and  $f_2 : \mathbb{R}^n \rightarrow [2^{nR}]$ , when the optimal decoder is used, it holds that

$$D^{\text{IP}} = \frac{1}{n} [\text{trace}(\Sigma_{e_U^{f_1}}) + \text{trace}(\Sigma_{e_U^{f_2}}) - \text{trace}(\Sigma_{e_U^{f_1}} \Sigma_{e_U^{f_2}})] \quad (56)$$

$$= \frac{1}{n} [\text{trace}(\Sigma_{e_U^{f_1}}) + \text{trace}(\Sigma_{e_U^{f_2}}) - \text{trace}(\Sigma_{e_U^{f_1}} \Sigma_{e_U^{f_2}})], \quad (57)$$

where the last equality follows since  $P = Q$ , and therefore  $U$  and  $V$  have the same distribution. By the Cauchy-Schwartz inequality,

$$\text{trace}(\Sigma_{e_U^{f_1}} \Sigma_{e_U^{f_2}}) = \sum_{i=1}^n \sum_{j=1}^n \Sigma_{e_U^{f_1}}(i, j) \Sigma_{e_U^{f_2}}(j, i) \quad (58)$$

$$\leq \sqrt{\sum_{i=1}^n \sum_{j=1}^n \Sigma_{e_U^{f_1}}^2(i, j) \sum_{i=1}^n \sum_{j=1}^n \Sigma_{e_U^{f_2}}^2(j, i)} \quad (59)$$

$$= \|\Sigma_{e_U^{f_1}}\|_F \|\Sigma_{e_U^{f_2}}\|_F. \quad (60)$$

Note that the upper bound above is tight if  $\Sigma_{e_U^{f_1}}$  is a scaled version of  $\Sigma_{e_U^{f_2}}$  (recall that  $\Sigma_{e_U^f} = \Sigma_{e_U^f}^\top$  since this is a covariance matrix), and in particular, this holds if  $f_1 = f_2 = f$ . Recall that for a positive semi-definite symmetric matrix  $A \in \mathbb{R}^n$  with eigenvalues  $\alpha = \text{eig}(A) = (\alpha_1, \dots, \alpha_n)$  it holds that

$$\text{trace}(A) = \sum_{i=1}^n \alpha_i = \|\alpha\|_1 \quad (61)$$

$$\|A\|_F = \sqrt{\sum_{i=1}^n \alpha_i^2} = \|\alpha\|_2. \quad (62)$$

Thus, combining (57) and (60), we obtain

$$D^{\text{IP}} \geq \frac{1}{n} [\|\lambda(f_1)\|_1 + \|\lambda(f_2)\|_1 - \|\lambda(f_1)\|_2 \cdot \|\lambda(f_2)\|_2], \quad (63)$$

with equality if  $f_1 = f_2$ . Set  $x_1 = \|\lambda(f_1)\|_1$ ,  $x_2 = \|\lambda(f_1)\|_2$ ,  $y_1 = \|\lambda(f_2)\|_1$  and  $y_2 = \|\lambda(f_2)\|_2$ . With this notation, we have

$$nD^{\text{IP}} \geq x_1 + y_1 - x_2 y_2 \quad (64)$$

$$= \frac{1}{2}(2x_1 - x_2^2) + \frac{1}{2}(2y_1 - y_2^2) + \frac{1}{2}(x_2 - y_2)^2 \quad (65)$$

$$\geq \frac{1}{2}(2x_1 - x_2^2) + \frac{1}{2}(2y_1 - y_2^2) \quad (66)$$

$$\geq \min \{2x_1 - x_2^2, 2y_1 - y_2^2\} \quad (67)$$

$$\geq \min_f [2\|\lambda(f)\|_1 - \|\lambda(f)\|_2^2]. \quad (68)$$

all inequalities are attained with equality if  $f_1 = f_2 = f^*$ , where  $f^*$  is a minimizer of (68). ■

The following Shannon-lower-bound-type lemma constrains the eigenvalues of an MSE matrix for estimating  $U$  from a  $2^{nR}$ -level quantizer  $f : \mathbb{R}^n \rightarrow [2^{nR}]$ .

*Lemma 2:* Assume (without loss of generality) that the distribution  $P$  has unit variance. Let  $f : \mathbb{R}^n \rightarrow [2^{nR}]$  be a  $2^{nR}$ -level quantizer, and define  $\lambda(f) = (\lambda_1, \dots, \lambda_n) \in [0, 1]^n$  as in (55). Then

$$\frac{1}{n} \sum_{i=1}^n \frac{1}{2} \log \frac{1}{\lambda_i} \leq R + D(P \|\mathcal{N}(0, 1)). \quad (69)$$

**Proof.** We may assume without loss of generality that  $h(P) > -\infty$ , as otherwise  $D(P \|\mathcal{N}(0, 1)) = \infty$  and the statement trivially holds. Let  $e_U^f = U - \mathbb{E}[U|f(U)]$ ,  $\Sigma_{e_U^f} = \mathbb{E}[e_U^f e_U^{f \top}]$ . Since the Gaussian distribution maximizes differential entropy under second moment constraints, we have that

$$h(U|f(U)) \leq \frac{1}{2} \log \det \left( (2\pi e) \Sigma_{e_U^f} \right) = n \cdot \frac{1}{n} \sum_{i=1}^n \frac{1}{2} \log(2\pi e \lambda_i). \quad (70)$$

Consequently,

$$nR \geq I(U; f(U)) = h(U) - h(U|f(U)) \geq h(U) - n \cdot \frac{1}{n} \sum_{i=1}^n \frac{1}{2} \log(2\pi e \lambda_i) \quad (71)$$

$$= h(\mathcal{N}^{\otimes n}(0, 1)) - n \cdot \frac{1}{n} \sum_{i=1}^n \frac{1}{2} \log(2\pi e \lambda_i) + h(P^{\otimes n}) - h(\mathcal{N}^{\otimes n}(0, 1)) \quad (72)$$

$$= n \left( \frac{1}{n} \sum_{i=1}^n \frac{1}{2} \log \frac{1}{\lambda_i} - D(P \|\mathcal{N}(0, 1)) \right), \quad (73)$$

which yields the claimed result. ■

*Theorem 6:* Assume (without loss of generality) that the distribution  $P$  has unit variance. Then,

$$D^{\text{IP},*}(R, P) \geq \Gamma(R + D(P \|\mathcal{N}(0, 1))), \quad (74)$$

where  $\Gamma(R)$  is defined in (1).

**Proof.** Let  $f : \mathbb{R}^n \rightarrow [2^{nR}]$  be a  $2^{nR}$ -level quantizer, and define  $\lambda(f) = (\lambda_1, \dots, \lambda_n) \in [0, 1]^n$  as in (55). Denote by  $K = K_f$  the uniform distribution over (the multiset)  $\lambda(f)$ . By Lemma 1, we have that

$$D_n^{\text{IP},*}(R, P) = \inf_f \mathbb{E}_{\lambda \sim K_f} \phi(\lambda) = \inf_f \mathbb{E}_{\lambda \sim K_f} \phi \left( 2^{-2R_{\mathcal{N}}(\lambda)} \right), \quad (75)$$

where  $R_{\mathcal{N}}(\lambda) = \frac{1}{2} \log \frac{1}{\lambda}$ . Denote  $\Gamma_1(R) = \phi(2^{-2R})$ . In Appendix A we show that

$$\text{convex envelope of } \Gamma_1(R) = \Gamma(R). \quad (76)$$

It therefore follows that

$$D_n^{\text{IP},*}(R, P) = \inf_f \mathbb{E}_{\lambda \sim K_f} \Gamma_1(R_{\mathcal{N}}(\lambda)) \quad (77)$$

$$\geq \inf_f \mathbb{E}_{\lambda \sim K_f} \Gamma(R_{\mathcal{N}}(\lambda)) \quad (78)$$

$$\geq \inf_f \Gamma(\mathbb{E}_{\lambda \sim K_f} R_{\mathcal{N}}(\lambda)) \quad (79)$$

$$\geq \Gamma(R + D(P \parallel \mathcal{N}(0, 1))), \quad (80)$$

where we have used Lemma 2 in the last inequality. ■

### C. The Symmetric Gaussian case

Combining Theorem 5 and Theorem 6, we obtain a complete characterization for the Gaussian case.

*Theorem 7:*

$$D^{\text{IP},*}(R, \mathcal{N}(0, 1)) = \Gamma(R) = \begin{cases} 1 - (1 - \phi(2^{-2R^*})) \frac{R}{R^*} & R < R^* \\ \phi(2^{-2R}) & R \geq R^* \end{cases}. \quad (81)$$

**Proof.** The upper bound follows from applying Theorem 5 with  $\alpha = \min\{R/R^*, 1\}$ , and recalling that  $D_{\mathcal{N}(0,1)}(R) = 2^{-2R}$ . The lower bound follows directly from Theorem 6. ■

## IV. COMPRESSION FOR MATRIX MULTIPLICATION

### A. Setup

Let  $A \in \mathbb{R}^{n \times a}$  be a matrix whose entries are drawn iid from the distribution  $P$  and  $B \in \mathbb{R}^{n \times b}$  be a matrix, statistically independent of  $A$ , whose entries are drawn iid from the distribution  $Q$ . We assume both  $P$  and  $Q$  are distributions with unit-variance. We consider the problem of quantizing  $A$  and  $B$  in order to compute their matrix multiplication  $A^\top B$ . In particular, an  $(n, a, b, R_1, R_2, D)$  code consists of mappings

$$f_1 : \mathbb{R}^{n \times a} \rightarrow [2^{naR_1}] \quad (82)$$

$$f_2 : \mathbb{R}^{n \times b} \rightarrow [2^{nbR_2}] \quad (83)$$

$$g : [2^{naR_1}] \times [2^{nbR_2}] \rightarrow \mathbb{R}^{a \times b}, \quad (84)$$

with

$$D = D^{\text{MM}} = \frac{1}{n \cdot a \cdot b} \mathbb{E} \|A^\top B - g(f_1(A), f_2(B))\|_F^2. \quad (85)$$

We define

$$D_{n,a,b}^{\text{MM},*}(R_1, R_2) = D_{n,a,b}^{\text{MM},*}(R_1, R_2, P, Q) = \inf \{D : \exists(n, a, b, R_1, R_2, D) - \text{code}\}. \quad (86)$$

We further define the asymptotic function

$$D_{a,b}^{\text{MM}}(R_1, R_2) = D_{a,b}^{\text{MM}}(R_1, R_2, P, Q) = \limsup_{n \rightarrow \infty} D_{n,a,b}^*(R_1, R_2), \quad (87)$$

### B. Basic Properties and Bounds

Denote  $W_A = f_1(A)$  and  $W_B = f_2(B)$  and further denote  $\hat{A} = \mathbb{E}[A|W_A]$  and  $\hat{B} = \mathbb{E}[B|W_B]$ . Define  $\Sigma_A = \mathbb{E}[(A - \hat{A})(A - \hat{A})^\top] \in \mathbb{R}^{n \times n}$  and  $\bar{M}_A = \mathbb{E}[\hat{A}\hat{A}^\top] \in \mathbb{R}^{n \times n}$ . Similarly,  $\Sigma_B = \mathbb{E}[(B - \hat{B})(B - \hat{B})^\top] \in \mathbb{R}^{n \times n}$  and  $\bar{M}_B = \mathbb{E}[\hat{B}\hat{B}^\top] \in \mathbb{R}^{n \times n}$ . As in the scalar case, we still have the identities:

$$\Sigma_A + \bar{M}_A = aI_n \quad (88)$$

$$\Sigma_B + \bar{M}_B = bI_n. \quad (89)$$

The next theorem generalizes the basic bounds we derived above for the inner product case, to the matrix multiplication case. The proofs are similar to the statements above, and are therefore omitted.

*Theorem 8:* The following hold:

1) For fixed  $f_1, f_2$ , the optimal choice for  $g$  is  $g^*(W_A, W_B) = \hat{A}^\top \hat{B}$ , and the distortion is given by

$$\begin{aligned} D^{\text{MM}} &= \frac{1}{n \cdot a \cdot b} [\text{trace}(\Sigma_A \bar{M}_B) + \text{trace}(\Sigma_B \bar{M}_A) + \text{trace}(\Sigma_A \Sigma_B)] \\ &= \frac{1}{n} \left[ \frac{1}{a} \text{trace}(\Sigma_A) + \frac{1}{b} \text{trace}(\Sigma_B) - \frac{1}{a \cdot b} \text{trace}(\Sigma_A \Sigma_B) \right]. \end{aligned}$$

2) The oracle lower bound (taking  $\hat{B} = B$  or  $\hat{A} = A$ ) gives

$$D^{\text{MM}} \geq \max \left\{ \frac{1}{n \cdot a} \text{trace} \Sigma_A, \frac{1}{n \cdot b} \text{trace} \Sigma_B \right\},$$

and consequently for any  $n \geq 1$

$$D_{n,a,b}^{\text{MM},*}(R_1, R_2, P, Q) \geq \max \{D_P(R_1), D_Q(R_2)\},$$

and in particular

$$D^{\text{MM}}(R_1, R_2, P, Q) \geq \max \{D_P(R_1), D_Q(R_2)\}.$$

3) For the symmetric case, where  $R_1 = R_2 = R$  and  $P = Q$ , we have

$$D_{a,b}^{\text{MM}}(R, P) \leq \min_{0 \leq \alpha \leq 1} (1 - \alpha) + \alpha \cdot \phi \left( D_P \left( \frac{R}{\alpha} \right) \right)$$

This is asymptotically attained by quantizing only the first  $\alpha n$  coordinates of each column of  $A$  and each column of  $B$

4) For the symmetric case, where  $R_1 = R_2 = R$  and  $P = Q$ , for any  $n \geq 1$  we have

$$\begin{aligned} D_{n,a,b}^{\text{MM},*}(R, P) &\geq \frac{1}{n} \min \left\{ \inf_{f_a} [2\|\lambda(f_a)\|_1 - \|\lambda(f_a)\|_2^2], \inf_{f_b} [2\|\lambda(f_b)\|_1 - \|\lambda(f_b)\|_2^2] \right\} \\ &= \frac{1}{n} \min \left\{ \inf_{f_a} \sum_{i=1}^n \phi(\lambda_i(f_a)), \inf_{f_b} \sum_{i=1}^n \phi(\lambda_i(f_b)) \right\}, \end{aligned} \quad (90)$$

where the infima runs over all encoders  $f_a : \mathbb{R}^{n \times a} \rightarrow [2^{naR}]$ ,  $f_b : \mathbb{R}^{n \times b} \rightarrow [2^{nbR}]$ , and

$$\lambda(f_a) = \text{eig} \left( \frac{1}{a} \Sigma_{e_{f_a}^a} \right), \quad \lambda(f_b) = \text{eig} \left( \frac{1}{b} \Sigma_{e_{f_b}^b} \right) \quad (91)$$

where  $e_{f_a}^a = A - \mathbb{E}[A|f_a(A)]$ ,  $\Sigma_{e_{f_a}^a} = \mathbb{E}[e_{f_a}^a e_{f_a}^{a,\top}]$ , and  $e_{f_b}^b = B - \mathbb{E}[B|f_b(B)]$ ,  $\Sigma_{e_{f_b}^b} = \mathbb{E}[e_{f_b}^b e_{f_b}^{b,\top}]$ .

### C. Maximum Entropy Matrices

The fact that the Gaussian distribution maximizes the differential entropy of a vector, under second moment constraints, played a pivotal role in the derivation of our bounds for inner product quantization. For matrix multiplication quantization, the following lemma will play a similar role.

*Lemma 3:* Let  $M \in \mathbb{R}^{n \times a}$  be a random matrix with  $\mathbb{E}[M] = 0$ , and  $\mathbb{E}[MM^\top] = \Sigma$ . Then

$$h(M) \geq \frac{a}{2} \log \det \left( 2\pi e \frac{1}{a} \Sigma \right) = a \cdot \sum_{i=1}^n \frac{1}{2} \log(2\pi e \lambda_i), \quad (92)$$

where  $\lambda = \text{eig} \left( \frac{1}{a} \Sigma \right)$ , and this is attained with equality if the columns of  $M$  are independent  $\mathcal{N} \left( 0, \frac{1}{a} \Sigma \right)$  random vectors.

**Proof.** Write  $M = [m_1 | m_2 | \dots | m_a]$ , where  $m_1, \dots, m_a$  are zero-mean random vectors in  $\mathbb{R}^n$ . Denote the marginal distribution of  $m_i$  by  $P_i$ . Let  $\Sigma_i = \mathbb{E}[m_i m_i^\top]$ , and recall that

$$\Sigma = \mathbb{E}[MM^\top] = \sum_{i=1}^a \mathbb{E}[m_i m_i^\top] = \sum_{i=1}^a \Sigma_i. \quad (93)$$

We further have that

$$h(M) = h(m_1, \dots, m_a) \leq \sum_{i=1}^a h(m_i) \leq a \cdot h \left( \frac{1}{a} \sum P_i \right), \quad (94)$$

where we have used sub-additivity and concavity of differential entropy in the inequalities above. Noting that the covariance matrix corresponding to the distribution  $\frac{1}{a} \sum_{i=1}^a P_i$  is  $\frac{1}{a} \sum_{i=1}^a \Sigma_i = \frac{1}{a} \Sigma$ , we have

$$h(M) \leq a \cdot h\left(\mathcal{N}\left(0, \frac{1}{a} \Sigma\right)\right) = \frac{a}{2} \log \det\left(2\pi e \frac{1}{a} \Sigma\right). \quad (95)$$

All inequalities are attained with equality when  $m_i \stackrel{iid}{\sim} \mathcal{N}\left(0, \frac{1}{a} \Sigma\right)$ , for  $i = 1, \dots, a$ . ■

This immediately gives the following generalization of Lemma 2

*Lemma 4:* Assume (without loss of generality) that the distribution  $P$  has unit variance. Let  $f_a : \mathbb{R}^{n \times a} \rightarrow [2^{naR}]$  be a  $2^{naR}$ -level quantizer, and define  $\lambda(f_a) = (\lambda_1, \dots, \lambda_n) \in [0, 1]^n$  as in (91). Then

$$\frac{1}{n} \sum_{i=1}^n \frac{1}{2} \log \frac{1}{\lambda_i} \leq R + D(P \|\mathcal{N}(0, 1)). \quad (96)$$

**Proof.** Without loss of generality, we may assume  $h(P) \geq -\infty$ , as otherwise  $D(P \|\mathcal{N}(0, 1)) = \infty$  and the statement is trivial. Let  $e_A^{f_a} = A - \mathbb{E}[A|f_a(A)]$ ,  $\Sigma_{e_A^{f_a}} = \mathbb{E}[e_A^{f_a} e_A^{f_a, \top}]$ . By Lemma 3, we have that

$$h(A|f_a(A)) \leq \frac{a}{2} \log \det\left((2\pi e) \frac{1}{a} \Sigma_{e_A^{f_a}}\right) = na \cdot \frac{1}{n} \sum_{i=1}^n \frac{1}{2} \log(2\pi e \lambda_i). \quad (97)$$

Consequently,

$$naR \geq I(A; f_a(A)) = h(A) - h(A|f_a(A)) \geq h(A) - na \cdot \frac{1}{n} \sum_{i=1}^n \frac{1}{2} \log(2\pi e \lambda_i) \quad (98)$$

$$= h(\mathcal{N}^{\otimes na}(0, 1)) - na \cdot \frac{1}{n} \sum_{i=1}^n \frac{1}{2} \log(2\pi e \lambda_i) + h(P^{\otimes na}) - h(\mathcal{N}^{\otimes na}(0, 1)) \quad (99)$$

$$= na \left( \frac{1}{n} \sum_{i=1}^n \frac{1}{2} \log \frac{1}{\lambda_i} - D(P \|\mathcal{N}(0, 1)) \right), \quad (100)$$

which yields the claimed result. ■

#### D. Fundamental Limits

Using Theorem 8 and Lemma 4, we prove the following result for the symmetric matrix multiplication case.

*Theorem 9:* Assume (without loss of generality) that the distribution  $P$  has unit variance and finite differential entropy. Then, for any  $n \geq 1$

$$D_{n,a,b}^{\text{MM},*}(R, P) \geq \Gamma(R + D(P \|\mathcal{N}(0, 1))), \quad (101)$$

where  $\Gamma(R)$  is defined in (1).

**Proof.** Let  $f_a : \mathbb{R}^{n \times a} \rightarrow [2^{naR}]$  be a  $2^{naR}$ -level quantizer, and define  $\lambda(f_a) = (\lambda_1, \dots, \lambda_n) \in [0, 1]^n$  as in (91). Denote by  $K = K_{f_a}$  the uniform distribution over (the multiset)  $\lambda(f_a)$ , and  $R_{\mathcal{N}}(\lambda) = \frac{1}{2} \log \frac{1}{\lambda}$ , as in the proof of Theorem 6, we have that

$$\mathbb{E}_{\lambda \sim K_{f_a}} \phi(\lambda) = \mathbb{E}_{\lambda \sim \Gamma_{f_a}} \phi\left(2^{-2R_{\mathcal{N}}(\lambda)}\right). \quad (102)$$

Recalling from the proof of Theorem 6 that the  $\Gamma_1(R) = \phi(2^{-2R}) \geq \Gamma(R)$  and the function  $R \mapsto \Gamma(R)$  is convex and non-increasing, it follows that

$$\mathbb{E}_{\lambda \sim K_{f_a}} \phi\left(2^{-2R_{\mathcal{N}}(\lambda)}\right) \geq \mathbb{E}_{\lambda \sim K_f} \Gamma(R_{\mathcal{N}}(\lambda)) \quad (103)$$

$$\geq \Gamma(\mathbb{E}_{\lambda \sim K_f} R_{\mathcal{N}}(\lambda)) \quad (104)$$

$$\geq \Gamma(R + D(P \|\mathcal{N}(0, 1))), \quad (105)$$

where we have used Lemma 4 in the last inequality. Thus,

$$\frac{1}{n} \sum_{i=1}^n \phi(\lambda_i(f_a)) = \mathbb{E}_{\lambda \sim K_{f_a}} \phi(\lambda) \geq \Gamma(R + D(P \|\mathcal{N}(0, 1))). \quad (106)$$



Similarly, for any  $f_b : \mathbb{R}^{n \times b} \rightarrow [2^{nbR}]$  we have

$$\frac{1}{n} \sum_{i=1}^n \phi(\lambda_i(f_b)) = \mathbb{E}_{\lambda \sim K_{f_b}} \phi(\lambda) \geq \Gamma(R + D(P \|\mathcal{N}(0, 1))). \quad (107)$$

Thus, by (90) in Theorem 8, for any  $n \geq 1$

$$D_{n,a,b}^{\text{MM}}(R, P) \geq \min \left\{ \min_{f_a} \frac{1}{n} \sum_{i=1}^n \phi(\lambda_i(f_a)), \min_{f_b} \frac{1}{n} \sum_{i=1}^n \phi(\lambda_i(f_b)) \right\} \quad (108)$$

$$\geq \Gamma(R + D(P \|\mathcal{N}(0, 1))), \quad (109)$$

as claimed. ■

**Proof of Theorem 2.** Part 1 follows immediately from Theorem 9. Part 2 follows from part 2 of Theorem 8, and recalling that  $D_{\mathcal{N}(0,1)}(R) = 2^{-2R}$ . ■

### E. The Symmetric Gaussian case

Combining Theorem 8 and Theorem 9, we obtain a complete characterization for the Gaussian case.

*Theorem 10:*

$$D_{a,b}^{\text{MM}}(R, \mathcal{N}(0, 1)) = \Gamma(R). \quad (110)$$

**Proof.** The upper bound follows applying Part 3 of Theorem 8 with  $\alpha = \min\{R/R^*, 1\}$ , and recalling that  $D_{\mathcal{N}(0,1)}(R) = 2^{-2R}$ . The lower bound follows directly from Theorem 9. ■

## V. LATTICE QUANTIZATION SCHEME FOR MATRIX MULTIPLICATION OF ARBITRARY MATRICES

Our theoretical analysis in Sections II-IV assumed the entries in the vectors/matrices to be multiplied are drawn iid from some known distribution. In this section, we drop this assumption, and, building on the observations from the analysis above, develop a robust scheme for compression for matrix multiplication. Our scheme is designed to attain the optimal distortion in the case where  $A$  and  $B$  have iid Gaussian entries, but the error it attains for arbitrary matrices can also be upper bounded.

We first develop encoders  $f_1, f_2 : \mathbb{R}^n \rightarrow [2^{nR}]$  and a decoder  $g : [2^{nR}] \times [2^{nR}] \rightarrow \mathbb{R}$  for estimating the inner product of  $U, V \in \sqrt{n}\mathbb{S}^{n-1}$  where  $\mathbb{S}^{n-1} = \{x \in \mathbb{R}^n : \|x\| = 1\}$  is the unit sphere. We then show how these encoders and decoder can be leveraged for compression for matrix multiplication. Let  $O_n(\mathbb{R})$  be the orthogonal group, consisting of all orthonormal matrices in  $\mathbb{R}^{n \times n}$ . It will be useful to analyze the performance of  $f_1, f_2, g$  with respect to the following distribution on  $U, V$ .

*Definition 1 ( $\rho$ -correlated spherically uniform random vectors):* Let  $S = [S_1 | S_2 | \dots | S_n] \sim \text{Uniform}(O_n(\mathbb{R}))$  be a random matrix uniformly distributed over the group of orthogonal matrices in  $\mathbb{R}^{n \times n}$  (that is,  $S$  is drawn from the Haar measure on  $O_n(\mathbb{R})$ ). We say that the random vectors  $U \in \mathbb{R}^n$  and  $V \in \mathbb{R}^n$  are  $\rho$ -correlated spherically uniform random vectors if  $U = \sqrt{n}S_1$ ,  $Z = \sqrt{n}S_2$  and

$$V = \rho U + \sqrt{1 - \rho^2} Z. \quad (111)$$

*Theorem 11:* For any  $\varepsilon > 0$  and sufficiently large  $n$

- 1) There exist encoders  $f_1, f_2 : \mathbb{R}^n \rightarrow [2^{nR}]$  and a decoder  $g : [2^{nR}] \times [2^{nR}] \rightarrow \mathbb{R}$ , such that if  $U, V$  are  $\rho$ -correlated spherically uniform

$$\mathbb{E}(U^\top V - g(f_1(U), f_2(V)))^2 < n(\Gamma(R) - \Gamma^2(R) + \varepsilon) + \rho^2 n^2 (\Gamma^2(R) + \varepsilon), \quad (112)$$

for every  $0 \leq \rho \leq 1$ , where  $\Gamma(R)$  is defined in (1).

- 2) There exists an encoder  $f : \mathbb{R}^n \rightarrow [2^{nR}]$  and a decoder  $g : [2^{nR}] \times \mathbb{R}^n \rightarrow \mathbb{R}$ , such that if  $U, V$  are  $\rho$ -correlated spherically uniform

$$\mathbb{E}(U^\top V - g(f(U), V))^2 < n(2^{-2R} - 2^{-4R} + \varepsilon) + \rho^2 n^2 (2^{-4R} + \varepsilon), \quad (113)$$

for every  $0 \leq \rho \leq 1$ .

The proof of Theorem 11 is based on the nested lattice coding scheme described in Subsection V-A1, and its performance analysis in Subsection V-A2 and Subsection V-B. A simple variation of the scheme described in

Subsection V-A1, with essentially the same (though much simplified) performance analysis gives the following result.

*Theorem 12:* For any  $\varepsilon > 0$  and sufficiently large  $n$

- 1) There exist encoders  $f_1, f_2 : \mathbb{R}^n \rightarrow [2^{nR}]$  and a decoder  $g : [2^{nR}] \times [2^{nR}] \rightarrow \mathbb{R}$ , such that if  $U, V$  are  $\rho$ -correlated spherically uniform

$$\mathbb{E}(U^\top V - g(f_1(U), f_2(V)))^2 < n \left( \frac{2 \cdot 2^{2R} - 1}{(2^{2R} - 1)^2} + \varepsilon \right) \quad (114)$$

for every  $0 \leq \rho \leq 1$ .

- 2) There exists an encoder  $f : \mathbb{R}^n \rightarrow [2^{nR}]$  and a decoder  $g : [2^{nR}] \times \mathbb{R}^n \rightarrow \mathbb{R}$ , such that if  $U, V$  are  $\rho$ -correlated spherically uniform

$$\mathbb{E}(U^\top V - g(f(U), V))^2 < n \left( \frac{1}{2^{2R} - 1} + \varepsilon \right) \quad (115)$$

for every  $0 \leq \rho \leq 1$ .

With these two theorems, can now easily prove Theorem 1 and Theorem 3.

**Proof of Theorem 1 and of Theorem 3.** We only prove part 1 of the two theorems. The proofs for part 2 are nearly identical, and we highlight the required modifications in the end of the proof.

Let  $f_1, f_2, g$  be the encoders and decoder from either Theorem 11 or Theorem 12. Based on those  $f_1, f_2, g$ , we propose the following rate- $R$  quantization scheme for quantization of matrices  $A \in \mathbb{R}^{n \times a}$  and  $B \in \mathbb{R}^{n \times b}$  in order to estimate  $C = A^\top B$ :

- 1) Quantize the numbers  $\|a_1\|, \dots, \|a_a\|$  (respectively  $\|b_1\|, \dots, \|b_b\|$ ) in very high resolution so that we can ignore their quantization error in the analysis that follows. In particular, we assume the quantized values satisfy  $\|\widehat{a}_i\| \in (1 \pm n^{-3})\|a_i\|$ ,  $\|\widehat{b}_j\| \in (1 \pm n^{-3})\|b_j\|$  for all  $i = 1, \dots, a$ ,  $j = 1, \dots, b$ . Those are just  $a$  (respectively  $b$ ) numbers, so the cost of this description is negligible next to the  $naR$  (respectively,  $nbR$ ) bits budget;<sup>5</sup>
- 2) Draw  $S \sim \text{Uniform}(O_n(\mathbb{R}))$  at both encoders (using common randomness), and compute  $\tilde{A} = [\tilde{a}_1 | \dots | \tilde{a}_a] = SA$ , and  $\tilde{B} = [\tilde{b}_1 | \dots | \tilde{b}_b] = SB$ ;
- 3) Let

$$U_i = \sqrt{n} \frac{\tilde{a}_i}{\|\tilde{a}_i\|} = \sqrt{n} S \frac{a_i}{\|a_i\|}, \quad i = 1, \dots, a \quad (116)$$

$$V_j = \sqrt{n} \frac{\tilde{b}_j}{\|\tilde{b}_j\|} = \sqrt{n} S \frac{b_j}{\|b_j\|}, \quad j = 1, \dots, b. \quad (117)$$

Apply  $f_1 : \mathbb{R}^n \rightarrow [2^{nR}]$  on  $U_i$ , for  $i = 1, \dots, a$ , and  $f_2 : \mathbb{R}^n \rightarrow [2^{nR}]$  on  $V_j$ , for  $j = 1, \dots, b$ .

- 4) Use  $g : [2^{nR}] \times [2^{nR}] \rightarrow \mathbb{R}$ , to estimate each entry of  $C = A^\top B$  as<sup>6</sup>

$$\hat{C}_{ij} = \frac{\|\widehat{a}_i\| \|\widehat{b}_j\|}{n} g(f_1(U_i), f_2(V_j)), \quad i = 1, \dots, a, \quad j = 1, \dots, b. \quad (118)$$

<sup>5</sup>Let us describe a quantizer for  $\|a_i\|$  that uses  $O(\log n)$  bits and produces a reconstruction  $\|\widehat{a}_i\| \in (1 \pm n^{-3})\|a_i\|$ . Note that if  $A$  has “bounded entries”, then  $\|a_i\| \in \{0\} \cup [2^{-2000}, \sqrt{n}2^{2000}]$ . To quantize it with multiplicative error at most  $\delta = n^{-3}$  we construct a grid of  $m + 1$  points:  $\{0, 2^{-2000}, (1 + \delta)2^{-2000}, (1 + \delta)^2 \cdot 2^{-2000}, \dots, (1 + \delta)^m \cdot 2^{-2000}\}$ , where

$$\log m = \log \left( 4000 + \frac{1}{2} \log n \right) - \log \log(1 + \delta) = O(\log n)$$

and quantize  $\|a_i\|$  to the nearest point in the grid (recall that  $\frac{1}{1+\delta} \geq 1 - \delta$ ). Since  $\log m = o(n)$  for any finite  $\delta$ , the cost of describing  $\|a_1\|, \dots, \|a_a\|$  is indeed negligible, and same for  $\|b_1\|, \dots, \|b_b\|$ .

<sup>6</sup>Strictly speaking, we only have access to the quantized versions  $\|\widehat{a}_i\| \in (1 \pm n^{-3})\|a_i\|$  and  $\|\widehat{b}_j\| \in (1 \pm n^{-3})\|b_j\|$ . Thus,

$$\hat{C}_{ij} = \frac{\|\widehat{a}_i\| \|\widehat{b}_j\|}{n} g(f_1(U_i), f_2(V_j)) = \frac{\|\widehat{a}_i\| \|\widehat{b}_j\|}{n} g(f_1(U_i), f_2(V_j)) \pm \frac{\|\widehat{a}_i\| \|\widehat{b}_j\|}{n} \frac{3g(f_1(U_i), f_2(V_j))}{n^3}.$$

Since  $|U_i^\top V_j| \leq n$ , we can assume without loss of generality that  $|g(f_1(U_i), f_2(V_j))| \leq n$  and consequently

$$\hat{C}_{ij} = \frac{\|\widehat{a}_i\| \|\widehat{b}_j\|}{n} g(f_1(U_i), f_2(V_j)) = \frac{\|\widehat{a}_i\| \|\widehat{b}_j\|}{n} (g(f_1(U_i), f_2(V_j)) + O(n^{-2})),$$

so that the  $O(n^{-2})$  can be ignored without affecting our result.

To analyze the mean squared error  $\mathbb{E}(C_{ij} - \hat{C}_{i,j})^2$ , first note that

$$C_{ij} = a_i^\top b_j = a_i^\top S^\top S b_j = \tilde{a}_i^\top \tilde{b}_j = \frac{\|a_i\| \|b_j\|}{n} U_i^\top V_j. \quad (119)$$

Let  $\rho_{ij} = \frac{a_i^\top b_j}{\|a_i\| \|b_j\|}$ . We claim that  $U_i, V_j$  are  $\rho_{ij}$ -correlated spherically uniform random vectors. To see this, note that due to the random rotation matrix  $S$ , we may assume without loss of generality that

$$\frac{a_i}{\|a_i\|} = [1|0|0|\dots|0]^\top, \quad (120)$$

$$\frac{b_j}{\|b_j\|} = [\rho_{ij}|\sqrt{1-\rho_{ij}^2}|0|\dots|0]^\top, \quad (121)$$

and this assumption will have no affect on the joint distribution of  $U_i, V_j$ . Writing  $S = [S_1|S_2|\dots|S_n]$ , we therefore have that  $U_i = \sqrt{n}S_1$  and  $V_j = \rho_{ij}U_i + \sqrt{1-\rho_{ij}^2}Z$ , with  $Z = \sqrt{n}S_2$ . Thus, if  $f_1, f_2, g$  are the encoders and decoder from Theorem 11, we therefore have that

$$\mathbb{E}(U_i^\top V_j - g(f_1(U_i), f_2(V_j)))^2 < n(\Gamma(R) - \Gamma^2(R) + \varepsilon) + \rho_{ij}^2 n^2 (\Gamma^2(R) + \varepsilon). \quad (122)$$

Consequently,

$$\begin{aligned} \mathbb{E}(C_{ij} - \hat{C}_{i,j})^2 &= \mathbb{E} \left( \frac{\|a_i\| \|b_j\|}{n} (U_i^\top V_j - g(f_1(U_i), f_2(V_j))) \right)^2 \\ &< \frac{\|a_i\|^2 \|b_j\|^2}{n} (\Gamma(R) - \Gamma^2(R) + \varepsilon) + C_{ij}^2 (\Gamma^2(R) + \varepsilon). \end{aligned} \quad (123)$$

The proof of Theorem 1 is complete, by noting that

$$\begin{aligned} \frac{1}{n} \sum_{i,j} \|a_i\|^2 \|b_j\|^2 &= \frac{1}{n} \sum_{i=1}^a \|a_i\|^2 \sum_{j=1}^b \|b_j\|^2 = \frac{\|A\|_F^2 \|B\|_F^2}{n}, \\ \sum_{i,j} C_{ij}^2 &= \|C\|_F^2. \end{aligned} \quad (124)$$

If  $f_1, f_2, g$  are the encoders and decoder from Theorem 12, we have that

$$\mathbb{E}(U_i^\top V_j - g(f_1(U_i), f_2(V_j)))^2 < n \left( \frac{2 \cdot 2^{2R} - 1}{(2^{2R} - 1)^2} + \varepsilon \right), \quad (125)$$

and consequently,

$$\begin{aligned} \mathbb{E}(C_{ij} - \hat{C}_{i,j})^2 &= \mathbb{E} \left( \frac{\|a_i\| \|b_j\|}{n} (U_i^\top V_j - g(f_1(U_i), f_2(V_j))) \right)^2 \\ &< \frac{\|a_i\|^2 \|b_j\|^2}{n} \left( \frac{2 \cdot 2^{2R} - 1}{(2^{2R} - 1)^2} + \varepsilon \right), \end{aligned} \quad (126)$$

which completes the proof of Theorem 3.

The proofs for part 2 are identical with  $f(U_i) = f_1(U_i)$  and with  $f_2(V_j) = V_j$  (which is possible, since the decoder has access to  $V_j$ ). ■

Next we prove Theorem 11. We first consider the case of  $R \geq R^*$  and provide the details of the coding scheme, that is specify the encoders  $f_1, f_2$  and the decoder  $g$  in Subsection V-A1, and analyze its performance in Subsection V-A2. In Subsection V-A3 we treat the case of  $R < R^*$ , where time-sharing is needed. The one sided-quantization case, where only  $U$  is quantized and the decoder has access to  $V$  is handled in Subsection V-B. The proof of Theorem 12 is given in Subsection V-C. It uses the same scheme described in Subsection V-A1 with the only difference being that the parameter  $\alpha$  from the definition of the coding scheme is chosen as  $\alpha = 1$ , rather than  $\alpha = \sqrt{1-D}$  as in the proof of 11. The analysis is exactly the same as in Subsection V-A2, except for the different choice of  $\alpha$ , and time-sharing is not used at all.

### A. Proof of Theorem 11, Part 1

1) *Dithered Nested Lattice Quantization for Inner Product*: It will be convenient to denote the dimension by  $d$  rather than  $n$ . Thus, we assume in this and in the next subsection that  $U$  and  $V$  are  $\rho$ -correlated spherically uniform random vectors in  $\mathbb{R}^d$  rather than in  $\mathbb{R}^n$ . The reason is that we will later use time-sharing, and describe only the first  $d \leq n$  coordinates of  $U$  and  $V$  using the scheme below.

Let  $\Lambda_c \subset \Lambda_f$  be a pair of nested lattices in  $\mathbb{R}^d$ . Assume that  $|\Lambda_f/\Lambda_c| = 2^{dR}$ . See [37] for basic lattice definitions. Denote by  $\mathcal{V}_c$  the Voronoi region of  $\Lambda_c$  and by  $\mathcal{V}_f$  the Voronoi region of  $\Lambda_f$ . Let  $\tilde{Z}_1, \tilde{Z}_2 \sim \text{Uniform}(\mathcal{V}_f)$  be statistically independent dither vectors. Let  $0 < D < 1$ , and assume that

$$\sigma^2(\Lambda_f) = \frac{1}{d} \mathbb{E} \|\tilde{Z}_1\|^2 = D. \quad (127)$$

This is without loss of generality, as we can always scale both lattices  $\Lambda_c \subset \Lambda_f$  by the same factor  $\beta > 0$ , so that the lattice  $\Lambda_c$  will satisfy this constraint. We denote the nearest neighbor quantizer with respect to the lattice  $\Lambda_f$ , applied on  $x \in \mathbb{R}^d$ , as

$$Q_{\Lambda_f}(x) = \operatorname{argmin}_{\lambda \in \Lambda_f} \|x - \lambda\|, \quad (128)$$

where ties are broken arbitrarily, but in systematic manner. The modulo operation with respect to the lattice  $\Lambda_f$ , is defined in this paper as

$$[x] \bmod \Lambda_f = x - Q_{\Lambda_f}(x). \quad (129)$$

Note that  $[x] \bmod \Lambda_f \in \mathcal{V}_f$ . The operations  $Q_{\Lambda_c}(x)$  and  $[x] \bmod \Lambda_c$  are defined similarly.

Let

$$\alpha = \sqrt{1 - D}. \quad (130)$$

Our encoders  $f_1, f_2 : \mathbb{R}^d \rightarrow [2^{dR}]$  compute

$$\tilde{U} = \left[ Q_{\Lambda_f} \left( \alpha U + \tilde{Z}_1 \right) \right] \bmod \Lambda_c \quad (131)$$

$$\tilde{V} = \left[ Q_{\Lambda_f} \left( \alpha V + \tilde{Z}_2 \right) \right] \bmod \Lambda_c, \quad (132)$$

and each of them maps the result to  $dR$  bits (which is possible since  $|\Lambda_f/\Lambda_c| = 2^{dR}$ ).

The decoder  $g(f_1(U), f_2(V))$  computes

$$\hat{U} = \alpha \left( \left[ \tilde{U} - \tilde{Z}_1 \right] \bmod \Lambda_c \right) \quad (133)$$

$$\hat{V} = \alpha \left( \left[ \tilde{V} - \tilde{Z}_2 \right] \bmod \Lambda_c \right), \quad (134)$$

and estimates the inner product as  $\hat{U}^\top \hat{V}$ .

2) *Analysis*: We now analyze the performance of this scheme. First, note that

$$\begin{aligned} \left[ \tilde{U} - \tilde{Z}_1 \right] \bmod \Lambda_c &= \left[ \left[ Q_{\Lambda_f} \left( \alpha U + \tilde{Z}_1 \right) \right] \bmod \Lambda_c - \tilde{Z}_1 \right] \bmod \Lambda_c \\ &= \left[ Q_{\Lambda_f} \left( \alpha U + \tilde{Z}_1 \right) - \tilde{Z}_1 \right] \bmod \Lambda_c \\ &= \left[ \alpha U + \left( Q_{\Lambda_f} \left( \alpha U + \tilde{Z}_1 \right) - (\alpha U + \tilde{Z}_1) \right) \right] \bmod \Lambda_c \\ &= [\alpha U + Z_1] \bmod \Lambda_c, \end{aligned} \quad (135)$$

where

$$Z_1 = Q_{\Lambda_f} \left( \alpha U + \tilde{Z}_1 \right) - (\alpha U + \tilde{Z}_1) \quad (136)$$

is uniform over  $\mathcal{V}_1$  and statistically independent of  $U$  (and everything else), by the Crypto Lemma [37], [38]. By the same analysis, we obtain

$$\left[ \tilde{V} - \tilde{Z}_2 \right] \bmod \Lambda_c = [\alpha V + Z_2] \bmod \Lambda_c \quad (137)$$

where  $Z_2 \sim \text{Uniform}(\mathcal{V}_f)$  is statistically independent of  $V$  (and everything else). For the remainder of the analysis we will make the following assumption.

*Assumption 1:* It holds that

$$[\alpha U + Z_1] \bmod \Lambda_c = \alpha U + Z_1 \iff \alpha U + Z_1 \in \mathcal{V}_c, \quad (138)$$

$$[\alpha V + Z_2] \bmod \Lambda_c = \alpha V + Z_2 \iff \alpha V + Z_2 \in \mathcal{V}_c. \quad (139)$$

We discuss the conditions under which this assumption holds with high probability in the sequel.

Under Assumption 1, we have

$$\begin{aligned} \hat{U} &= \alpha(\alpha U + Z_1), & e_U &= U - \hat{U} = (1 - \alpha^2)U - \alpha Z_1, \\ \hat{V} &= \alpha(\alpha V + Z_2), & e_V &= V - \hat{V} = (1 - \alpha^2)V - \alpha Z_2. \end{aligned} \quad (140)$$

Thus, the error in estimating the inner product is

$$\begin{aligned} e &= U^\top V - \hat{U}^\top \hat{V} = (\hat{U} + e_U)^\top (\hat{V} + e_V) - \hat{U}^\top \hat{V} = \hat{U}^\top e_V + \hat{V}^\top e_U + e_U^\top e_V \\ &= \alpha [(\alpha U + Z_1)^\top ((1 - \alpha^2)V - \alpha Z_2)] \\ &\quad + \alpha [(\alpha V + Z_2)^\top ((1 - \alpha^2)U - \alpha Z_1)] \\ &\quad + ((1 - \alpha^2)U - \alpha Z_1)^\top ((1 - \alpha^2)V - \alpha Z_2) \\ &= \alpha [2\alpha(1 - \alpha^2)U^\top V + (1 - \alpha^2)(U^\top Z_2 + V^\top Z_1) - \alpha^2(U^\top Z_2 + V^\top Z_1) - 2\alpha Z_1^\top Z_2] \\ &\quad + (1 - \alpha^2)^2 U^\top V - \alpha(1 - \alpha^2)(U^\top Z_2 + V^\top Z_1) + \alpha^2 Z_1^\top Z_2 \\ &= (1 - \alpha^2)(1 + \alpha^2)U^\top V - \alpha^3(U^\top Z_2 + V^\top Z_1) - \alpha^2 Z_1^\top Z_2. \end{aligned} \quad (141)$$

Recalling (111), we obtain

$$e = (1 - \alpha^4) \left( \rho \|U\|^2 + \sqrt{1 - \rho^2} U^\top Z \right) - \alpha^3 (U^\top Z_2 + \rho U^\top Z_1 + \sqrt{1 - \rho^2} Z^\top Z_1) - \alpha^2 Z_1^\top Z_2. \quad (142)$$

We note that the estimator is biased and

$$\mathbb{E}(e) = (1 - \alpha^4) \rho \mathbb{E}\|U\|^2 = d\rho\phi(D), \quad (143)$$

where in the last equality we have used

$$(1 - \alpha^4) = 1 - (1 - D)^2 = 2D(1 - D) = \phi(D). \quad (144)$$

Let us now compute  $\mathbb{E}(e^2)$ . Since  $U^\top Z = 0$ , and  $\|U\|^2 = d$  with probability 1, and since  $(U, Z_1, Z_2)$  are statistically independent, and  $(Z, Z_1, Z_2)$  are statistically independent, we have that

$$\mathbb{E}(e^2) = (1 - \alpha^4)^2 \rho^2 d^2 + \alpha^6 (dD + \rho^2 dD + (1 - \rho^2) dD) + \alpha^4 dD^2 \quad (145)$$

$$= d [\rho^2 d(1 - \alpha^4)^2 + \alpha^4 D(2\alpha^2 + D)] \quad (146)$$

$$= d [\rho^2 d(1 - \alpha^4)^2 + \alpha^4 D(2 - D)] \quad (147)$$

$$= d [\rho^2 d\phi^2(D) + (1 - \phi(D))\phi(D)] \quad (148)$$

$$= d [(\phi(D) - \phi^2(D)) + \rho^2 d\phi^2(D)]. \quad (149)$$

Validity of Assumption 1: We now turn to discuss the validity of Assumption 1. To this end, we will need a few more definitions. Following [9], we say that a sequence (in  $d$ ) of random vectors  $X^{(d)} \in \mathbb{R}^d$  is *semi norm-ergodic* if for any  $\varepsilon > 0$  it holds that

$$\lim_{d \rightarrow \infty} \Pr(\|X^{(d)}\|^2 > (1 + \varepsilon)\mathbb{E}\|X^{(d)}\|^2) = 0. \quad (150)$$

For a lattice  $\Lambda \subset \mathbb{R}^d$  we define the *effective radius*  $r_{\text{eff}}(\Lambda)$  as the radius of an  $\ell_2$  ball whose volume is equal to  $\text{covol}(\Lambda) = \text{Vol}(\mathcal{V}) = |\det(G)|$ , where  $\mathcal{V}$  is the Voronoi region of  $\Lambda$  and  $G \in \mathbb{R}^{d \times d}$  is a generating matrix for  $\Lambda$  (that is,  $\Lambda = G\mathbb{Z}^d$ ). In particular,  $r_{\text{eff}}(\Lambda)$  satisfies the equation  $\text{covol}(\Lambda) = \text{Vol}(r_{\text{eff}}(\Lambda)\mathcal{B}) = V_d \cdot r_{\text{eff}}^d(\Lambda)$ , where  $\mathcal{B} = \{x \in \mathbb{R}^d : \|x\| \leq 1\}$  is the unit ball, whose volume is  $V_d = \frac{\pi^{d/2}}{\Gamma(\frac{d}{2} + 1)}$ . We say that a sequence of nested lattice pairs  $\Lambda_c \subset \Lambda_f$  in  $\mathbb{R}^d$  is “good” if for any  $\varepsilon > 0$ :

1)

$$\lim_{d \rightarrow \infty} \frac{d\sigma^2(\Lambda_f)}{r_{\text{eff}}^2(\Lambda_f)} = 1, \quad \lim_{d \rightarrow \infty} \frac{d\sigma^2(\Lambda_c)}{r_{\text{eff}}^2(\Lambda_c)} = 1 \quad (151)$$

2) For any semi norm-ergodic sequence of random vectors with  $\frac{1}{d}\mathbb{E}\|X^{(d)}\|^2 < \sigma^2(\Lambda_f)$  it holds that

$$\lim_{d \rightarrow \infty} \Pr(X^{(d)} \notin \Lambda_f) = 0, \quad (152)$$

and for any semi norm-ergodic sequence of random vectors with  $\frac{1}{d}\mathbb{E}\|Y^{(d)}\|^2 < \sigma^2(\Lambda_c)$  it holds that

$$\lim_{d \rightarrow \infty} \Pr(Y^{(d)} \notin \Lambda_c) = 0, \quad (153)$$

By [9, Theorem 2], for any  $0 < D < 1$  and  $R > 0$ , there exists a sequence of “good” nested lattice pairs  $\Lambda_c \subset \Lambda_f$ , with  $\sigma^2(\Lambda_f) = D$  and  $\frac{1}{d} \log |\Lambda_f/\Lambda_c| = R$ . Note that the fact that the sequence is “good”, further implies that

$$\frac{\sigma^2(\Lambda_c)}{\sigma^2(\Lambda_f)} \rightarrow \frac{r_{\text{eff}}^2(\Lambda_c)}{r_{\text{eff}}^2(\Lambda_f)} = \frac{\text{covol}(\Lambda_c)^{2/d}}{\text{covol}(\Lambda_f)^{2/d}} = |\Lambda_f/\Lambda_c|^{2/d} = 2^{2R}. \quad (154)$$

The sequence of random vectors  $U = U^{(d)}$  has deterministic norm, and is therefore trivially semi norm-ergodic. For a sequence of “good” nested lattices, it therefore follows by [9, Theorem 3] that  $\alpha U + Z_1$  is semi-norm ergodic as well. Thus, as long as  $\sigma^2(\Lambda_c) > \frac{1}{d}\mathbb{E}\|\alpha U + Z_1\|^2 = (1-D) + D = 1$ , we have that  $\Pr(\alpha U + Z_1 \notin \mathcal{V}_c)$  is as small as desired (for large enough  $d$ ). Thus, if  $R > \frac{1}{2} \log(1/D)$ , or equivalently  $D > 2^{-2R}$ , we can find lattices  $\Lambda_c \subset \Lambda_f$  for which Assumption 1 holds with high probability. Since for  $R \geq R^*$  and  $D = 2^{-2R}$  we have that  $\phi(D) = \phi(2^{-2R}) = \Gamma(R)$ , taking  $d = n$ , this establishes Theorem 11 for  $R \geq R^*$ .

We note that in the rare cases where Assumption 1 fails to hold, we can scale both lattices  $\Lambda_c \subset \Lambda_f$  by some  $\beta > 1$ , for which Assumption 1 does hold. Such scaling results in increasing  $D$  to  $\beta^2 D$ . Note that the encoders  $f_1, f_2$  can easily check whether or not Assumption 1 holds. Reporting to the decoder that such  $\beta$  was used has negligible effect on the coding rate. Since these cases are so rare, the effect on the expected distortion is also negligible. We therefore omit these details from the analysis. In Section VI we discuss a practical implementation of the scheme above, where such modulo-errors cannot be made rare, and the mechanism for increasing  $\beta$  and reporting it is described in detail.

3) *Time Sharing*: For the case  $R < R^*$ , we need to combine the scheme above with time-sharing. Set  $\kappa = R/R^*$ , and  $d = \kappa n$ . We assume  $\kappa n$  is an integer to avoid uninteresting, and insignificant, rounding effects. We use a sequence of “good” nested lattices  $\Lambda_c \subset \Lambda_f$  in  $\mathbb{R}^d$  with  $\sigma^2(\Lambda_f) = D > 2^{-2R^*}$ , such that  $\sigma^2(\Lambda_c) \rightarrow \sigma^2(\Lambda_f) \cdot 2^{2R/\kappa} > 1$ . Therefore, for  $d$  large enough it is possible to take  $D$  as close as desired to  $2^{-2R^*}$  without violating Assumption 1. Let  $\mathcal{M} = [d] = \{1, \dots, d\}$  and  $\bar{\mathcal{M}} = [n] \setminus [d] = \{d+1, \dots, n\}$ . We use the nested lattice code described above, for  $U_{\mathcal{M}}$  and  $V_{\mathcal{M}}$  and denote  $e_{\mathcal{M}} = U_{\mathcal{M}}^{\top} V_{\mathcal{M}}^{\top} - \hat{U}_{\mathcal{M}}^{\top} \hat{V}_{\mathcal{M}}$ . We do not allocate rate for the description of the remaining coordinates  $\bar{\mathcal{M}}$ , and simply estimate the inner product  $U_{\bar{\mathcal{M}}}^{\top} V_{\bar{\mathcal{M}}}$  as  $\hat{U}_{\bar{\mathcal{M}}}^{\top} \hat{V}_{\bar{\mathcal{M}}} = 0$ . Further denote  $e_{\bar{\mathcal{M}}} = U_{\bar{\mathcal{M}}}^{\top} V_{\bar{\mathcal{M}}} - \hat{U}_{\bar{\mathcal{M}}}^{\top} \hat{V}_{\bar{\mathcal{M}}} = U_{\bar{\mathcal{M}}}^{\top} V_{\bar{\mathcal{M}}}$ .

Crucially, in the  $\rho$ -correlated spherically uniform case on  $(\mathbb{R}^n)^2$ , projection of  $U$  and  $V$  to  $d < n$  coordinates changes the basic structure of their joint distribution. In particular, denoting the projection of  $U$  on the first  $d$  coordinates by  $U_{[d]}$ , and similarly for  $Z_{[d]}$ , the variables  $\|U_{[d]}\|$  and  $U_{[d]}^{\top} Z_{[d]}$  are no longer deterministic if  $d < n$ . It holds that (using the fact that  $\mathbb{E}(U_i^2) = 1$ )

$$\mathbb{E}\|U_{[d]}\|^2 = d. \quad (155)$$

By symmetry, we also have

$$\mathbb{E}(U_{[d]}^{\top} Z_{[d]}) = 0. \quad (156)$$

Finally, using the fact that  $\mathbb{E}(U_i^4) = \frac{3n}{n+2}$  [39], in Appendix B, we prove that

$$\mathbb{E}\|U_{[d]}\|^4 = \frac{n}{n+2} d(d+2). \quad (157)$$

and that

$$\mathbb{E}(U_{[d]}^{\top} Z_{[d]})^2 = \frac{n}{n+2} d \cdot \frac{n-d}{n-1}. \quad (158)$$

Using the error expression (142), we have

$$e_{\mathcal{M}} = (1 - \alpha^4) \left( \rho \|U_{[d]}\|^2 + \sqrt{1 - \rho^2} U_{[d]}^\top Z_{[d]} \right) - \alpha^3 (U_{[d]}^\top Z_2 + \rho U_{[d]}^\top Z_1 + \sqrt{1 - \rho^2} Z_{[d]}^\top Z_1) - \alpha^2 Z_1^\top Z_2. \quad (159)$$

Consequently,

$$\mathbb{E}(e_{\mathcal{M}}^2) = (1 - \alpha^4)^2 \left( \rho^2 \mathbb{E}\|U_{[d]}\|^4 + (1 - \rho^2) \mathbb{E}(U_{[d]}^\top Z_{[d]})^2 \right) + \alpha^6 \cdot 2Dd + \alpha^4 D^2 d. \quad (160)$$

Since  $\frac{n}{(n+2)(n-1)} \leq \frac{1}{n}$  for all  $n > 1$ , we have  $\mathbb{E}(U_{[d]}^\top Z_{[d]})^2 \leq \frac{d(n-d)}{n}$ . We also have that  $\mathbb{E}\|U_{[d]}\|^4 = d^2 \left( 1 + \frac{2(n-d)}{d(n+2)} \right)$ . Thus, for fixed  $\kappa$  and  $n$  large enough,  $\mathbb{E}\|U_{[d]}\|^4 < d^2(1 + o(n^{-1}))$ . In the sequel, we neglect the  $o(n^{-1})$  correction term, as it can be absorbed in the  $+\varepsilon$  terms in (112). Recalling that  $d = \kappa n$ , and substituting in (160), we obtain

$$\mathbb{E}(e_{\mathcal{M}}^2) \leq (1 - \alpha^4)^2 n (\rho^2 n \cdot \kappa^2 + (1 - \rho^2) \kappa(1 - \kappa)) + n\kappa\alpha^6 2D + n\kappa\alpha^4 D^2 \quad (161)$$

$$= n\kappa (\rho^2 \phi^2(D) [n\kappa - (1 - \kappa)] + \phi(D) - \kappa \phi^2(D)) \quad (162)$$

$$\leq n\kappa (\rho^2 n\kappa \phi^2(D) + \phi(D) - \kappa \phi^2(D)). \quad (163)$$

In similar fashion, we can show that

$$\mathbb{E}(e_{\bar{\mathcal{M}}}^2) \leq \rho^2 n^2 (1 - \kappa)^2 + n\kappa(1 - \kappa). \quad (164)$$

Finally, we show in Appendix B that

$$\mathbb{E}(e_{\mathcal{M}} e_{\bar{\mathcal{M}}}) \leq \phi(D) n\kappa(1 - \kappa) (\rho^2 n - 1). \quad (165)$$

Thus,

$$\mathbb{E}(e^2) = \mathbb{E}(e_{\mathcal{M}} + e_{\bar{\mathcal{M}}})^2 = \mathbb{E}(e_{\mathcal{M}}^2) + \mathbb{E}(e_{\bar{\mathcal{M}}}^2) + 2\mathbb{E}(e_{\mathcal{M}} e_{\bar{\mathcal{M}}}) \quad (166)$$

$$\leq n\kappa (\rho^2 n\kappa \phi^2(D) + \phi(D) - \kappa \phi^2(D)) + \rho^2 n^2 (1 - \kappa)^2 + n\kappa(1 - \kappa) + 2\phi(D) n\kappa(1 - \kappa) (\rho^2 n - 1) \quad (167)$$

$$= n \cdot \rho^2 n (\kappa \phi(D) + (1 - \kappa))^2 + n \cdot \left( (\kappa \phi(D) + (1 - \kappa)) - (\kappa \phi(D) + (1 - \kappa))^2 \right) \quad (168)$$

$$\leq n (\Gamma(R) - \Gamma^2(R) + \varepsilon) + \rho^2 n^2 (\Gamma^2(R) + \varepsilon), \quad (169)$$

where the last inequality follows since we can take  $D$  arbitrarily close to  $2^{-2R^*}$  without violating Assumption 1. This establishes (112).

### B. Proof of Theorem 11, Part 2

For the second part of Theorem 11 we use  $f = f_1$  in order to encode  $U$ , where  $f_1$  is the encoder introduced in Section V-A1, with  $d = n$  (no time-sharing). We set  $\hat{U}$  as in (133), and  $g(f(U), V) = \hat{U}^\top V$ . The analysis proceeds as in Section V-A2, but with  $\hat{V} = V$ , such that  $e_V = 0$ . We therefore have that

$$e = U^\top V - \hat{U}^\top V = (U - \hat{U})^\top V = ((1 - \alpha^2)U - \alpha Z_1)^\top V = (1 - \alpha^2)U^\top V - \alpha Z_1^\top V. \quad (170)$$

Recalling that  $\alpha = \sqrt{1 - D}$  and that  $U^\top V = \sqrt{n}\rho$ , we obtain

$$e = n\rho D - \alpha Z_1^\top V. \quad (171)$$

Recalling further that  $Z_1$  and  $V$  are statistically independent, and that  $\mathbb{E}\|Z_1\|^2 = nD$ , and  $\mathbb{E}[VV^\top] = I_n$ , we obtain

$$\mathbb{E}(e^2) = n^2 \rho^2 D^2 + \alpha^2 nD = nD(1 - D) + \rho^2 n^2 D^2. \quad (172)$$

Choosing  $D$  arbitrarily close to  $2^{-2R}$ , which is possible without violating (138) from Assumption 1 for ‘‘good’’ nested lattice pair  $\Lambda_c \subset \Lambda_f$  (by the arguments given in Section V-A2) completes the proof.

### C. Proof of Theorem 12

For the proof of part 1, we use the scheme described in Subsection V-A1 with  $d = n$  (no time-sharing), but with  $\alpha = 1$  (instead of  $\alpha = \sqrt{1-D}$ ). We take  $D$  to be slightly greater than  $\frac{1}{2^{2R}-1}$  (more accurately, we think of  $D$  as a sequence in  $n$ , approaching  $\frac{1}{2^{2R}-1}$  from above). By [9, Theorem 2], there exist a sequence of “good” nested lattices  $\Lambda_c \subset \Lambda_f$  with  $\sigma^2(\Lambda_f) = D$  and  $R = \frac{1}{d} \log |\Lambda_f/\Lambda_c|$ . By (154), for a sequence of “good” nested lattices we have

$$\sigma^2(\Lambda_c) \rightarrow \sigma^2(\Lambda_f)2^{2R} \rightarrow 1 + D. \quad (173)$$

Since  $U + Z_1$  is semi norm-ergodic with  $\frac{1}{n} \mathbb{E} \|U + Z_1\|^2 = 1 + D$ , we have that  $[U + Z_1] \bmod \Lambda_c = U + Z_1$  with high probability. Similarly  $[V + Z_2] \bmod \Lambda_c = V + Z_2$  holds with high probability, and consequently, Assumption 1 holds with high probability. Since here  $\alpha = 1$ , the error  $e$  as given in (142) has zero mean, and satisfies (145)  $\mathbb{E}(e^2) = n(2D + D^2)$ . Recalling that  $D \rightarrow \frac{1}{2^{2R}-1}$  this gives

$$\mathbb{E}(U^\top V - g(f_1(U), f_2(V)))^2 = \mathbb{E}(e^2) \leq n \left( \frac{2 \cdot 2^{2R} - 1}{(2^{2R} - 1)^2} + \varepsilon \right), \quad (174)$$

as claimed.

For the second part of Theorem 12 we still take  $\alpha = 1$  and use  $f = f_1$  in order to encode  $U$ , where  $f_1$  is the encoder introduced in Section V-A1, with  $d = n$  (no time-sharing). We set  $\hat{U}$  as in (133), and  $g(f(U), V) = \hat{U}^\top V$ . We use a sequence of “good” nested lattices with  $D \rightarrow \frac{1}{2^{2R}-1}$  as above, so that (138) from Assumption 1 holds with high probability, by the same considerations as above. The analysis proceeds as in Section V-A2, but with  $\hat{V} = V$ , such that  $e_V = 0$ . We therefore have that  $e = Z_1^\top V$  and

$$\mathbb{E}(e^2) = \mathbb{E} \|Z_1\|^2 = nD \leq n \left( \frac{1}{2^{2R} - 1} + \varepsilon \right), \quad (175)$$

as claimed.

## VI. PRACTICAL IMPLEMENTATION OF NESTED LATTICE QUANTIZERS

In the proof of Theorems 11-12 we used a pair of nested lattices  $\Lambda_c \subset \Lambda_f \subset \mathbb{R}^d$ , with  $|\Lambda_f/\Lambda_c| = 2^{dR}$ . Given such a pair of lattices in  $\mathbb{R}^d$ , in order to implement the coding scheme described above, we need to implement the following procedures:

- 1)  $Q_{\Lambda_f}(x) = \operatorname{argmin}_{\lambda_f \in \Lambda_f} \|x - \lambda_f\|$
- 2)  $Q_{\Lambda_c}(x) = \operatorname{argmin}_{\lambda_c \in \Lambda_c} \|x - \lambda_c\|$
- 3) Mapping from  $\Lambda_f/\Lambda_c$  to  $dR$  bits
- 4) Mapping from  $dR$  bits to the coset representatives  $\Lambda_f \cap \mathcal{V}_c$  of  $\Lambda_f/\Lambda_c$
- 5) Generating a random dither  $Z \sim \operatorname{Uniform}(\mathcal{V}_{\Lambda_f})$ , where  $\mathcal{V}_{\Lambda_f}$  is the Voronoi cell of  $\Lambda_f$

**Self-similar nested lattice codebooks:** Let  $\Lambda \subset \mathbb{R}^d$  be a lattice with generating matrix  $G \in \mathbb{R}^{d \times d}$ , such that  $\Lambda = G\mathbb{Z}^d$ . Assume that we have access to a procedure that implements the lattice quantizer  $Q_\Lambda(x)$  efficiently, and that there is some  $\kappa > 0$  such that  $\kappa\mathbb{Z}^d \subset \Lambda$ . The assumption that  $\mathbb{Z}^d$  is nested in  $\Lambda$  (up to scaling) is not very important, but also not restrictive, since the majority of lattices for which efficient lattice quantizers are known do satisfy it.

Using the lattice  $\Lambda$ , we can construct a pair of nested lattices  $\Lambda_c \subset \Lambda_f \subset \mathbb{R}^d$ , with  $|\Lambda_f/\Lambda_c| = 2^{dR}$ , that induce an efficiently implementable coding scheme. In particular, let  $\beta > 0$  and set  $\Lambda_f = \beta\Lambda$ ,  $\Lambda_c = q\Lambda_f = \beta \cdot q\Lambda$ , where  $q = 2^R$  is an integer. Algorithm 1 below provides the pseudo code for implementing  $f_1, f_2$  from Subsection V-A1 with  $\alpha = 1$  for such a nested lattice codebook. Note that the output `OverloadError` of Algorithm 1 specifies whether or not Assumption 1 holds, that is, whether or not a modulo error occurred. In order to implement the decoder  $g$  from Subsection V-A1 (again with  $\alpha = 1$ ), one implements (133) by applying Algorithm 2 on the output of  $f_1$ , implements (134) by applying Algorithm 2 on the output of  $f_2$ , and computes the inner product of the two vectors. In order to generate the random dithers  $\tilde{Z}_1, \tilde{Z}_2$ , one applies Algorithm 3.

**Choice of the parameter  $\beta$ :** Using this scheme, we have that

$$D = \sigma^2(\Lambda_f) = \beta^2 \sigma^2(\Lambda). \quad (176)$$



**Algorithm 1** NestedLatticeEncoder

**Inputs:** vector to be encoded  $x \in \mathbb{R}^{d'}$ , Lattice  $\Lambda \subset \mathbb{R}^{d'}$  with generating matrix  $G \in \mathbb{R}^{d' \times d'}$ , nesting ratio  $q \in \mathbb{N}$ , dither vector  $z \in \mathcal{V}_\Lambda \subset \mathbb{R}^{d'}$ , scaling factor  $\beta > 0$

**Outputs:**  $\text{Enc}(x) \in [q]^{d'}$  (can be represented using  $\lceil d' \log q \rceil$  bits), OverloadError that indicates if a modulo error occurred

$t \leftarrow Q_\Lambda \left( \frac{x}{\beta} + z \right)$   
 $y \leftarrow G^{-1}t$   
 $\text{Enc}(x) \leftarrow [y] \bmod q$  (elementwise modulo  $q$  reduction)

% check whether a modulo error occurred:

$\tilde{x} \leftarrow t - z$   
 $\lambda_c = q \cdot Q_\Lambda \left( \frac{\tilde{x}}{q} \right)$   
 OverloadError =  $\mathbb{1} \{ \lambda_c \neq 0 \}$

**Algorithm 2** NestedLatticeDecoder

**Inputs:** The encoding  $\text{Enc}(x) \in [q]^{d'}$  of  $x \in \mathbb{R}^{d'}$ , Lattice  $\Lambda \subset \mathbb{R}^{d'}$  with generating matrix  $G \in \mathbb{R}^{d' \times d'}$ , nesting ratio  $q \in \mathbb{N}$ , dither vector  $z \in \mathcal{V}_\Lambda \subset \mathbb{R}^{d'}$ , scaling factor  $\beta > 0$

**Outputs:**  $\hat{x} \in \mathbb{R}^{d'}$

$\tilde{y} \leftarrow G \cdot \text{Enc}(x) - z$   
 $\hat{x} \leftarrow \beta \left( \tilde{y} - q \cdot Q_\Lambda \left( \frac{\tilde{y}}{q} \right) \right)$

**Algorithm 3** GenerateRandomDither

**Inputs:** Lattice  $\Lambda \subset \mathbb{R}^{d'}$  and a number  $\kappa > 0$  such that  $\kappa \mathbb{Z}^{d'} \subset \Lambda$

**Outputs:**  $Z \sim \text{Uniform}(\mathcal{V}_\Lambda)$

$U \leftarrow \text{Uniform} \left( [0, \kappa]^{d'} \right)$   
 $Z \leftarrow U - Q_\Lambda(U)$

Thus, since the base lattice  $\Lambda$  is given, the parameter  $\beta$  controls  $D$ . We also have that

$$\sigma^2(\Lambda_c) = q^2 \sigma^2(\Lambda_f) = 2^{2R} D. \quad (177)$$

Assumption 1 is equivalent to  $U + Z_1 \in \mathcal{V}_c$  (and similarly,  $V + Z_2 \in \mathcal{V}_c$ ). If  $\Lambda$  is a high-dimensional lattice ( $d \gg 1$ ) that is “good” for quantization and for coding, this happens with high probability provided that  $1 + D = \frac{1}{d} \mathbb{E} \|U + Z_1\|^2 < \sigma^2(\Lambda_c) = 2^{2R} D$ , which is equivalent to  $D > D^*(R) = \frac{1}{2^{2R-1}}$ . In practice, we will usually work with a base lattice  $\Lambda$  whose second moment and coding goodness are sub-optimal. For this reason, we take  $D = \gamma D^*(R) = \frac{\gamma}{2^{2R-1}}$ , for some  $\gamma > 0$  (where  $\gamma$  is not necessarily close to 1), which is done by setting

$$\beta = \left( \frac{\gamma}{2^{2R-1}} \cdot \frac{1}{\sigma^2(\Lambda)} \right)^{1/2}. \quad (178)$$

**Avoiding modulo-error with negligible increase in rate:** Recall that Algorithm 1 also indicates, through the variable OverloadError, whether or not a modulo error occurred, that is, whether or not  $U + Z_1 \in \mathcal{V}_c$ . Whenever a modulo error does occur, one can increase the value of  $\gamma$  further to a large enough value, such that a modulo error does not occur with the new value, and inform the decoder on what value of  $\gamma$  was chosen. In practice, we may choose a bank of  $M$  values sorted in increasing order  $\gamma \in \{\gamma_1, \dots, \gamma_M\}$ . The encoder first uses  $\gamma_1$ . If OverloadError = 1 it tries again with  $\gamma_2$ , and keeps increasing  $\gamma$  to the next value until OverloadError = 0. If  $\gamma_1$  is chosen such that overload error is already not too common, and the values of  $\gamma_i$  increase sufficiently fast with  $i$ , say  $\gamma_i = i \cdot \gamma_1$ , the entropy of the first value of  $\gamma$  that returned OverloadError = 0 will be small. Since we only have to report this index to the decoder once for  $d$  symbols, the effect on the quantization rate is not significant.

Next, we develop a heuristic for choosing  $\gamma_1$ . Recall the definition of  $r_{\text{eff}}(\Lambda)$  from Section V-A2, and note that  $r_{\text{eff}}(\Lambda) = \left(\frac{\text{covol}(\Lambda)}{V_d}\right)^{1/d}$ . The normalized second moment (NSM) of a lattice  $\Lambda$  is defined as

$$N(\Lambda) = \frac{\sigma^2(\Lambda)}{(\text{covol}(\Lambda))^{2/d}} = \frac{\sigma^2(\Lambda)}{V_d^{2/d} r_{\text{eff}}^2(\Lambda)}. \quad (179)$$

If  $U + Z_1$  were Gaussian, the probability that it stays within  $\mathcal{V}_{\Lambda_c}$  would have been upper bounded by the probability that it stays within a ball with the same volume, that is, within a ball with radius  $r_{\text{eff}}(\Lambda_c)$ . Thus, we need  $r_{\text{eff}}^2(\Lambda_c)$  to be greater than  $\mathbb{E}\|U + Z_1\|^2$ . This corresponds to

$$1 < \frac{\frac{1}{d} r_{\text{eff}}^2(\Lambda_c)}{\frac{1}{d} \mathbb{E}\|U + Z_1\|^2} = \frac{1}{d} \frac{r_{\text{eff}}^2(\Lambda_c)}{\sigma^2(\Lambda_c)} \frac{\sigma^2(\Lambda_c)}{\frac{1}{d} \mathbb{E}\|U + Z_1\|^2} = \frac{1}{dV_d^{2/d} N(\Lambda)} \frac{2^{2R} D}{1 + D} = \frac{1}{dV_d^{2/d} N(\Lambda)} \frac{\gamma \cdot 2^{2R}}{2^{2R} + \gamma - 1} \approx \frac{\gamma}{dV_d^{2/d} N(\Lambda)}, \quad (180)$$

where the last approximation assumes that  $2^{2R} + \gamma - 1 \approx 2^{2R}$ . Thus, we will take

$$\gamma_1 \gtrsim dV^{2/d} N(\Lambda). \quad (181)$$

**Product lattices/Product quantization:** In order to use the self-similar nested lattice scheme described above, we need a base lattice  $\Lambda$  with an efficient nearest-neighbor decoder/lattice quantizer  $Q_\Lambda(x)$  and favorable quantization and coding properties. While it is easy to find (more accurately, to randomly draw) lattices in high-dimensions that are good for coding and quantization [9], [40], the task of finding such lattices that also admit an efficient nearest-neighbor decoder is notoriously difficult and is perhaps the holy grail of coding theory for the additive white Gaussian noise (AWGN) channel. A popular compromise between efficiency and “goodness”, is to use a *product lattice*, with a low-dimensional base lattice that is “pretty-good” for coding and quantization [41], [37].

Let  $d'$  be an integer that divides  $d$ , and  $\Lambda' \subset \mathbb{R}^{d'}$  be a lattice in  $\mathbb{R}^{d'}$ . We construct the lattice  $\Lambda \in \mathbb{R}^d$  as the product of  $L = d/d'$  copies of  $\Lambda'$ . Namely,

$$\Lambda = \underbrace{\Lambda' \times \cdots \times \Lambda'}_{L \text{ times}} = \Lambda'^{\otimes L} \quad (182)$$

The resulting self-similar nested lattices are also the product of  $L$  nested lattice pairs

$$\Lambda_c \subset \Lambda_f = (\beta_1 \cdot q\Lambda' \subset \beta_1\Lambda') \times \cdots \times (\beta_L \cdot q\Lambda' \subset \beta_L\Lambda'), \quad (183)$$

where we allow for different choices of  $\beta$  for each product to accommodate for the modulo-error control mechanism described above, but typically  $\beta_\ell = \left(\frac{\gamma_1}{2^{2R} - 1} \cdot \frac{1}{\sigma^2(\Lambda)}\right)^{1/2}$  since  $\gamma_1$  is already designed such that very few modulo-errors occur. Algorithm 1, Algorithm 2 and Algorithm 3 tensorize, and should be applied separately for each  $\ell = 1, \dots, L$  using the base lattice  $\Lambda' \subset \mathbb{R}^{d'}$  with generating matrix  $G' \in \mathbb{R}^{d' \times d'}$ . We also have that

$$\sigma^2(\Lambda) = \sigma^2(\Lambda') \cdot \frac{1}{L} \sum_{\ell=1}^L \beta_\ell^2. \quad (184)$$

Some lattices in small dimensions have excellent quantization and coding properties, as well as efficient nearest neighbor decoding algorithms. In particular,  $A_3 \cong D_3$  has the highest packing density among all lattices in  $\mathbb{R}^3$  [42],  $A_3^*$  has the smallest NSM among all lattices in  $\mathbb{R}^3$  [42] (only slightly smaller than that of  $A_3$ ),  $D_4$  has the highest packing density among all lattices in  $\mathbb{R}^4$  and lowest known NSM among all lattices in  $\mathbb{R}^4$  [42], [43], and  $E_8$  has the highest packing density (even among non-lattice packings) [44] and the smallest known NSM among all lattices in  $\mathbb{R}^8$  [42], [43]. All four lattices listed above, as well as many others from the  $A_n$ ,  $D_n$  and  $E_n$  families, admit a very fast lattice decoding algorithm [45]. Similarly, among all lattices in  $\mathbb{R}^{24}$ , the Leech lattice  $\Lambda_{24}$ , is the the best known quantizer [43], has the optimal packing density [46] (this is true even among all non-lattice packings [47]), and admits a pretty fast nearest neighbor decoding (or approximate nearest neighbor decoding) algorithms [48], [49], [50]. In addition, the second moment of all these lattices (and others) is calculated in [51] and reported also in [43, Table I]. Any one of those lattices is a good candidate for the base lattice  $\Lambda'$ . Another important advantage of these lattices is that they are all subsets of  $\mathbb{Z}^n$  up to scaling. Thus, when these lattices are used for quantization for matrix multiplication, and dithering is not applied, we can use integer multipliers (e.g., int8 tensor core in a GPU), rather than floating point multipliers, for multiplying the quantized matrices. The lattices of higher dimensions, and

in particular the Leech lattice, may yield better rate-distortion tradeoff than the lower-dimensional ones, but there are advantages to using lower-dimensional lattices in terms of efficiency. One of those is described next.

**Lookup tables:** Note that we decode  $\hat{U}_\ell \in \mathbb{R}^{d'}$  and  $\hat{V}_\ell \in \mathbb{R}^{d'}$  just to compute their inner product  $\hat{U}_\ell^\top \hat{V}_\ell$ . If we use the same dither vectors  $\tilde{Z}_1, \tilde{Z}_2 \in \mathcal{V}_{\Lambda'}$  for all  $\ell = 1, \dots, L$ , and the same value of  $\beta$ , namely,  $\beta_\ell^U = \beta_\ell^V = \beta$  for all  $\ell = 1, \dots, L$ , there are only  $q^{d'}$  values of  $\hat{U}_\ell^\top$  we can get, and only  $q^{d'}$  values of  $\hat{V}_\ell^\top$  we can get. Those do not depend on  $\ell$ . Thus, we can pre-compute all  $q^{2d'}$  possible values of  $\hat{U}_\ell^\top \hat{V}_\ell$  and store them in a lookup table (LUT). Then, instead of applying the decoder twice and computing the inner product, we simply fetch the result of the inner product from the LUT. If  $\beta_\ell^U \neq \beta$  or  $\beta_\ell^V \neq \beta$ , we simply multiply the value fetched from the LUT by  $\frac{\beta_\ell^U}{\beta} \cdot \frac{\beta_\ell^V}{\beta}$ . On some processors, using LUTs significantly speedup the decoding process, as it completely bypasses all lattice decoding operations, as well as all inner products. For approximate matrix multiplication  $A^\top B$  of  $A \in \mathbb{R}^{n \times a}$  and  $B \in \mathbb{R}^{n \times b}$  using the product nested lattice quantization scheme above, we need to perform  $a \cdot b \cdot (n/d')$  such operations, whereas the encoding only involves  $a(n/d') + b(n/d')$  lattice encoding operations. Thus, decoding is the computationally heavy procedure, and speeding it up will result in significant speedup of the total approximate matrix multiplication procedure. Using LUTs is therefore often highly advantageous. However, in order to have a very fast access time to the LUT, we would like it to “fit” in the highest levels of the cache, ideally in the L1 cache. This level has small capacity, which restricts the values of  $q^{2d'} = 2^{2Rd'}$ . Thus, we must keep  $Rd'$  small. Taking small  $R$  will typically not yield satisfactory resolution, so if LUTs are used, we are limited to using lattices  $\Lambda'$  of small dimensions. We note that for GPUs the LUT approach may not be attractive since the tensor core computes matrix multiplications extremely fast, while LUT probing is less efficient on this hardware. On CPUs on the other hand, the LUT approach can yield significant speed-up.

**Hadamard transform:** In Section V, (116-117) we proposed to multiply each column vector in  $A$  as well as each column vector in  $B$ , by a random projection matrix  $S$  drawn from the Haar distribution on  $O_n(\mathbb{R})$ . In general, the matrix  $S$  drawn from this distribution will have no structure, and calculating  $SA$  (respectively  $SB$ ) will require  $O(an^2)$  (respectively,  $O(bn^2)$ ) real-valued multiplication and summation operations. To significantly reduce the computational burden of this step, it was proposed in [5] (see also [6]) to restrict  $S$  to a certain class of orthogonal projection matrices: The randomized Hadamard transform. Here, we also follow this approach. In particular, we draw a vector  $T \sim \text{Uniform}(\{-1, 1\}^n)$ , and set  $K = \text{diag}(T)$ , that is,  $K$  is a diagonal matrix with  $K_{i,i} = T_i$ . We then set

$$S = \frac{1}{\sqrt{n}} HK, \quad (185)$$

where  $H \in \{-1, 1\}^{n \times n}$  is the Walsh-Hadamard matrix of dimension  $n$ . Here, we assumed that  $n$  is a power of 2, such that such a matrix exists. Otherwise, we can add rows of all zeros to both  $A$  and  $B$ , resulting in larger matrices  $A \in \mathbb{R}^{n' \times a}$  and  $B \in \mathbb{R}^{n' \times b}$ , with  $n' = 2^{\lceil \log_2(n) \rceil}$ . Note that in (116-117) we further scale the result by  $\sqrt{n}$ , so this cancels out the scaling by  $\frac{1}{\sqrt{n}}$  in (185). The gain for using the randomized Hadamard transform (185), is that its special fast-Fourier transform (FFT) structure allows to compute  $SA$  (respectively,  $SB$ ) using only  $O(an \log n)$  (respectively,  $O(bn \log n)$ ) additions and multiplications. Despite its simple implementation, the result of applying the randomized Hadamard transform on  $A$  (or  $B$ ) is quite similar to that of applying a “pure” random rotation on  $A$  (or  $B$ ) from various statistical perspectives [52], [53], [54].

**Representative numeric example:** To better illustrate how the building blocks above connect, we provide a numerical example. We have implemented a product nested lattice codebook, with  $\Lambda' = D_3$  (such that  $d' = 3$ ) as the base lattice. The lattice  $D_3$  consists of all vectors in  $\mathbb{Z}^3$  whose entries sum up to an even integer. In particular,  $2\mathbb{Z}^3 \subset D_3$ . The simple structure of  $D_3$  also gives rise to a very simple algorithm for computing  $Q_{D_3}(x)$  [45, Algorithm 2]. The lattice  $D_3$  has the highest packing density among all lattices in  $\mathbb{R}^3$  and its packing radius satisfies [42]  $r_{\text{pack}}(D_3)/r_{\text{eff}}(D_3) \approx (0.74)^{1/3} \approx 0.9045$ , such that its Voronoi region is quite close to a ball. We also have that  $\sigma^2(D_3) = \frac{3}{24}$ , so that  $N(D_3) \approx 0.0787$  (since  $\text{covol}(D_3) = 2$ ). This NSM is only slightly greater than the smallest NSM attained by any lattice in  $\mathbb{R}^3$ , which is  $N(A_3^*) \approx 0.0785$ .

We have used this base lattice with  $q = 6$  to construct a product nested lattice code as in (183). We used the same dither vectors  $\tilde{Z}_1, \tilde{Z}_2 \in \mathcal{V}_{D_3}$  for all  $\ell = 1, \dots, L$  (these vectors were drawn once at the beginning of the experiment). For this choice of  $d' = 3$  and  $q = 6$ , we can implement the decoder using a lookup table of size  $(q^3)^2 = 2^{6 \log_2 q} < 2^{15.6}$ . For constructing the LUT, we used the value  $\beta = 1$ . While for this choice of  $\beta$  all inner products between vectors in  $D_3$  are integer valued, because of the use of dithers, the entries in our LUT are

not integer-valued in general. We nevertheless rounded each of them to the nearest integer, and their range allows representing each entry in the LUT using an int8 variable. Consequently, the total size of the LUT is less than 64Kbyte, and it can be fully stored in the L1 cache of a modern processing unit.

For the lattice  $D_3$ , we have that the right-hand side of (181) evaluates to  $\approx 0.6139$ . We therefore choose  $\gamma_1 = 0.7$ , and set our bank of possible values of  $\gamma$  as  $\{i \cdot \gamma_1\}_{i=1}^9$ . The corresponding value of  $\beta$  is given by (178).

We drew two random matrices  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times n}$ , with all entries iid  $\mathcal{N}(0, 1)$ , where  $n = 3 \cdot 2^{11}$ . We used the product nested lattice codebook from (183) with  $L = 2^{11}$  for encoding each column of  $A$  (using the dither vector  $\tilde{Z}_1$ ) and for encoding each column of  $B$  (using the dither vector  $\tilde{Z}_2$ ). Since the iid Gaussian distribution is already rotation invariant, we have not implemented a random rotation. For each column, we also report the  $L$  values of  $\beta_i$  (equivalently  $\gamma_i$ ) used for each column. The (empirical) entropy of this random variable (that takes values in  $\beta_1 \cdot \{i\}_{i=1}^9$ ) for the choice  $\gamma_1 = 0.7$  was found to be around  $\approx 1.3$ bits. Since this value is only reported once for every  $d' = 3$  symbols (using entropy coding), its contribution to the coding rate is about 0.43 bits per symbol, such that the total rate of the coding scheme is  $R_{\text{eff}} \approx \log_2(6) + 0.43 \approx 3.015$  bits/symbol.

This approximate matrix multiplication algorithm attained  $\frac{1}{n^3} \|\widehat{A^\top B} - A^\top B\|_F^2 \approx 0.0593$ . Let  $e = \widehat{A^\top B} - A^\top B$ . The empirical distribution of the normalized approximation error  $e/\sqrt{n}$  (among the  $n^2$  entries) is plotted in Figure 1. Note that for  $R_{\text{eff}} = 3.015$ , Theorem 2 states that no scheme can attain distortion smaller than of  $\Gamma(R_{\text{eff}}) = 0.0304$  for  $A$  and  $B$  drawn as above, and Theorem 1 shows that this can be attained using high-dimensional lattices. Thus, our low-complexity implementation is not far of the optimal performance attained using optimal lattice codes. For comparison, we also evaluated the approximation error for a simple 3-bit scalar quantization scheme where each column  $a_i$  is normalized by  $\|a_i\|_\infty$  such that all its entries are in  $[-1, 1]$ , then each entry  $\tilde{a}_{i,t} = \frac{a_{i,t}}{\|a_i\|_\infty}$  is quantized to  $\frac{1}{4} \text{round}(4\tilde{a}_{i,t})$ , and in the end the quantized entries are rescaled again by  $\|a_i\|_\infty$ . The empirical error attained by the 3-bit scalar quantizer is  $\frac{1}{n^3} \|\widehat{A^\top B} - A^\top B\|_F^2 \approx 0.1668$ , about 3 times greater than the error attained using the  $D_3$ -based scheme with the same rate. The performance gap between the two scheme grows with  $n$ , as the random variable  $\|a_i\|_\infty$  concentrates around  $\sqrt{2 \ln n}$  for large  $n$ . Thus, the dynamic range for the scalar quantizer increases with  $n$ , which results in greater expected squared error.

## VII. OPEN PROBLEMS

One can interpret our Lemma 2 as follows: Let  $P = \mathcal{N}(0, 1)$  and  $U^n \sim P^{\otimes n}$ . Then for any random variable  $Y$  we have that

$$\sum_{i=1}^n R_P(\lambda_i) \leq I(U^n; Y), \quad (186)$$

where  $R_P(D)$  is the quadratic rate-distortion function for a source with distribution  $P$  and  $(\lambda_1, \dots, \lambda_n)$  are the eigenvalues of  $\text{Cov}(U^n|Y)$ . While Lemma 2 establishes (186) for the Gaussian distribution, we were not able to prove (186) for a general distribution, and we could neither find a counterexample. If (186) turns out to hold for any  $P$ , the proof of Theorem 6 could be easily extended to show that

$$D^{\text{IP},*}(R, P) = \text{convex envelope of } (\phi(D_P(R))), \quad (187)$$

where  $D_P(R)$  is the quadratic distortion-rate function for a source with distribution  $P$ . Thus, proving or disproving that (186) holds for all  $P$  is an interesting problem for future research.

In Theorem 1 we have shown the existence of encoders and decoder for quantization for matrix multiplication whose expected approximation error depends only on  $\|A\|_F^2 \cdot \|B\|_F^2$  and  $\|A^\top B\|_F^2$ , and is optimal for  $A$  and  $B$  whose entries are iid Gaussian. For iid Gaussian matrices we have that  $\frac{\mathbb{E}[\|A\|_F^2 \|B\|_F^2]/n}{\mathbb{E}[\|A^\top B\|_F^2]} = 1$  so that the two error terms in (3) are well-balanced. However, when the entries of  $A^\top B$  are large, that is when  $\frac{\|A\|_F^2 \|B\|_F^2/n}{\|A^\top B\|_F^2} = o(1)$ , the error term in (3) that involves  $\|A^\top B\|_F^2$  is the dominant one. On the other hand, in this case our bound from Theorem 3 gives much smaller error, despite the fact that it is not optimal for Gaussian matrices. In particular, the multiplicative error of the bound from Theorem 3 vanishes when  $\frac{\|A\|_F^2 \|B\|_F^2/n}{\|A^\top B\|_F^2} = o(1)$ . It is an interesting question for future research to understand whether for any  $R > 0$  there exist schemes that attain the optimal rate-distortion tradeoff for Gaussian iid matrices and at the same time attain a multiplicative error  $\frac{\mathbb{E}\|A^\top B - g(f_1(A), f_2(B))\|_F^2}{\|A^\top B\|_F^2}$  that vanishes when  $\frac{\|A\|_F^2 \|B\|_F^2/n}{\|A^\top B\|_F^2} = o(1)$ .

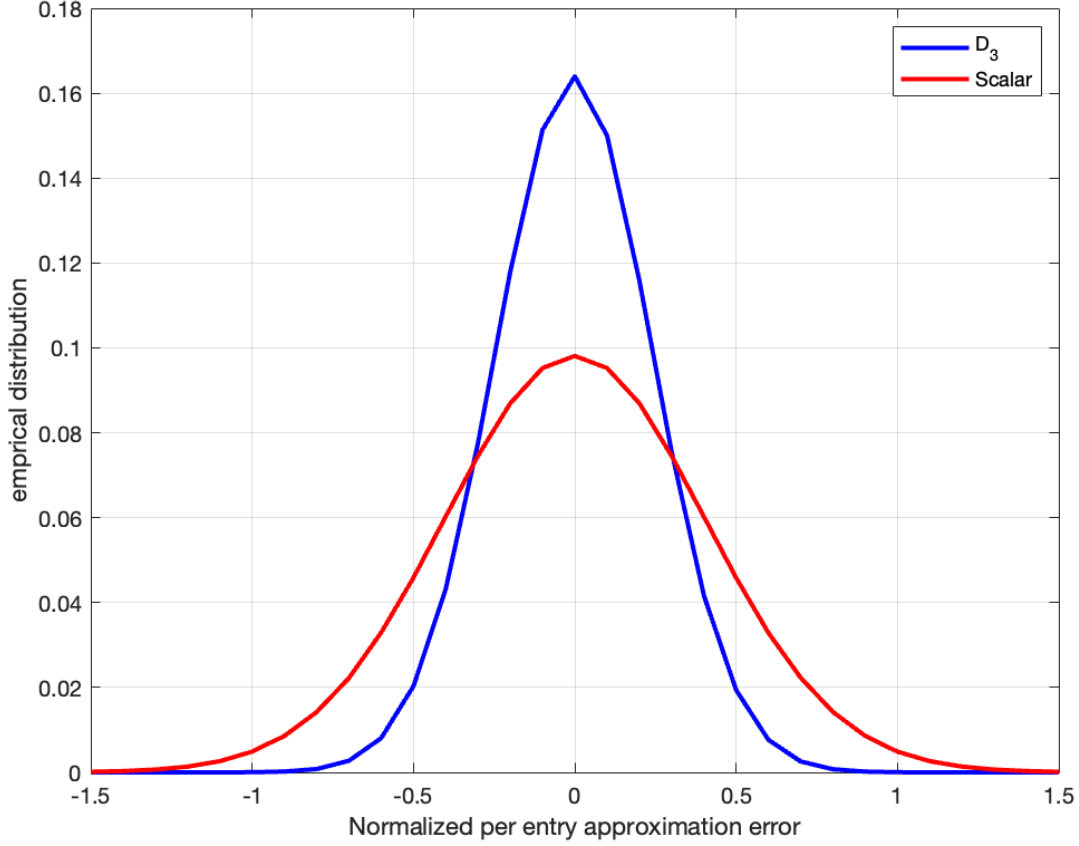


Fig. 1: The approximation error of the  $D_3$ -based product nested lattice coding scheme with  $q = 6$ , for random iid Gaussian matrices  $A, B \in \mathbb{R}^{n \times n}$ ,  $n = 3 \cdot 2^{11}$ . We plot the histogram of the entries of  $\frac{1}{\sqrt{n}}(\widehat{A^\top B} - A^\top B)$  in blue. For comparison, we also plot the histogram of the entries of  $\frac{1}{\sqrt{n}}(\widehat{A^\top B} - A^\top B)$  for a 3-bit scalar quantizer in red.

#### ACKNOWLEDGEMENTS

The authors thank Omri Weinstein (HUJI) for helping them navigate through the literature on approximate matrix multiplication and Yoon Kim (MIT) for explaining hardware and performance limitations of modern quantization algorithms in LLMs.

#### APPENDIX A CONVEX ENVELOPE OF $\Gamma_1(R)$

Recall that  $\phi(t) = 2t - t^2$  and

$$\Gamma_1(R) = \phi(2^{-2R}). \quad (188)$$

We show that the convex lower envelope of  $\Gamma_1(R)$  is  $\Gamma(R)$ . It is easy to verify that  $R \mapsto \Gamma_1(R)$  is decreasing, concave on  $[0, 1/2)$  and convex on  $(1/2, \infty)$ . Therefore, it will consist of a linear segment between  $(0, \Gamma_1(0) = 1)$  and  $(R^*, \Gamma_1(R^*))$  and agree with  $\Gamma_1(R)$  for  $R > R^*$ . The point  $R^* \geq 1/2$  is chosen such that the derivative of  $\Gamma(R)$  is smooth and non-decreasing. Thus, the convex envelope of  $\Gamma_1(R)$  is given by

$$\Gamma(R) = \begin{cases} \Gamma_1(R^*) + \Gamma_1'(R^*)(R - R^*) & R < R^* \\ \Gamma_1(R) & R \geq R^* \end{cases} \quad (189)$$

where  $R^*$  is chosen by requiring that  $\Gamma(0) = \Gamma_1(0) = 1$ , or in other words, that

$$\Gamma_1(R^*) - R^* \cdot \Gamma'_1(R^*) = 1. \quad (190)$$

Since  $\Gamma'_1(R^*) = -4 \ln 2 \cdot 2^{-2R^*} (1 - 2^{-2R^*})$  and we can express  $\Gamma_1(R^*)$  as  $\Gamma_1(R^*) = 2 \cdot 2^{-2R^*} (1 - 2^{-2R^*}) + 2^{-4R^*}$ , we have that (190) corresponds to

$$\begin{aligned} 2^{-4R^*} + 2 \cdot 2^{-2R^*} (1 - 2^{-2R^*}) (1 + 2 \ln 2R^*) &= 1 \\ \iff 2 \cdot 2^{-2R^*} (1 - 2^{-2R^*}) (1 + 2 \ln 2R^*) &= (1 - 2^{-2R^*}) (1 + 2^{-2R^*}) \\ \iff 2 \cdot 2^{-2R^*} (1 + 2 \ln 2R^*) &= 1 + 2^{-2R^*}, \\ \iff 1 + 4 \ln 2R^* &= 2^{2R^*}. \end{aligned} \quad (191)$$

$$\iff 1 + 4 \ln 2R^* = 2^{2R^*}. \quad (192)$$

## APPENDIX B

### PROJECTIONS OF RANDOM UNIFORM ORTHOGONAL VECTORS

Let  $S \sim \text{Uniform}(\text{O}_n(\mathbb{R}))$  and denote  $U = \sqrt{n}S_1$  and  $Z = \sqrt{n}S_2$ . To prove (157), we first note that, by symmetry

$$n^2 = \mathbb{E}\|U\|^4 = \mathbb{E}\left(\sum_{i=1}^n U_i^2\right)^2 = n\mathbb{E}(U_1^4) + n(n-1)\mathbb{E}(U_1^2U_2^2), \quad (193)$$

which implies

$$\mathbb{E}(U_1^2U_2^2) = \frac{n - \mathbb{E}(U_1^4)}{n-1} = \frac{n}{n+2}. \quad (194)$$

With this, we can write

$$\mathbb{E}\|U_{[d]}\|^4 = \mathbb{E}\left(\sum_{i=1}^d U_i^2\right)^2 = d\mathbb{E}(U_1^4) + d(d-1)\mathbb{E}(U_1^2U_2^2) = \frac{n}{n+2}d(d+2). \quad (195)$$

We move on to proving (158). We have that

$$\mathbb{E}(U_{[d]}^\top Z_{[d]})^2 = \mathbb{E}\left(\sum_{i=1}^d U_i Z_i\right)^2 = \sum_{i=1}^d \mathbb{E}(U_i^2 Z_i^2) + \sum_{j \neq i} \mathbb{E}(U_i U_j Z_i Z_j) = d\xi + d(d-1)\nu, \quad (196)$$

where

$$\xi = \mathbb{E}(U_1^2 Z_1^2), \quad \nu = \mathbb{E}(U_1 U_2 Z_1 Z_2), \quad (197)$$

and the last equality in (196) follows by symmetry. Taking  $d = n$ , we get that  $U_{[n]}^\top Z_{[n]} = U^\top Z = 0$  w.p. 1. Invoking (196) therefore gives

$$0 = n\xi + n(n-1)\nu \implies \nu = -\frac{\xi}{n-1}. \quad (198)$$

Substituting this into (196), we obtain

$$\mathbb{E}(U_{[d]}^\top Z_{[d]})^2 = d \left(1 - \frac{d-1}{n-1}\right) \xi = \frac{d(n-d)}{n-1} \xi. \quad (199)$$

In order to compute  $\xi$ , define  $e = U - Z$ . Note that the symmetry and orthogonality of  $U$  and  $Z$  implies that  $e \sim \text{Uniform}(\sqrt{2n}\mathbb{S}^{n-1})$ , where  $\mathbb{S}^{n-1}$  is the unit sphere in  $\mathbb{R}^n$ . It therefor follows that

$$\mathbb{E}(e_1^4) = 4\mathbb{E}(U_1^4). \quad (200)$$

On the other hand

$$\mathbb{E}(e_1^4) = \mathbb{E}(U_1 - Z_1)^4 = \sum_{i=0}^4 \binom{4}{i} \mathbb{E}(U_1^i Z_1^{4-i}) = \mathbb{E}(U_1^4) + \mathbb{E}(Z_1^4) + 6\mathbb{E}(U_1^2 Z_1^2) + 4\mathbb{E}(U_1 Z_1^3) + 4\mathbb{E}(U_1^3 Z_1). \quad (201)$$

By symmetry, we clearly have that  $\mathbb{E}(Z_1^4) = \mathbb{E}(U_1^4)$ . We claim that  $\mathbb{E}(U_1 Z_1^3) = 0$ . To see this, note that given  $Z$ , the distribution of  $U$  is invariant to negation (in other words  $p_{U|Z=z}(u) = p_{U|Z=z}(-u)$ ). By symmetry, this also implies that  $\mathbb{E}(U_1^3 Z_1) = 0$ . We therefore have that

$$4\mathbb{E}(U_1^4) = \mathbb{E}(e_1^4) = 2\mathbb{E}(U_1^4) + 6\xi \quad (202)$$

$$\implies \xi = \frac{\mathbb{E}(U_1^4)}{3} = \frac{n}{n+2}. \quad (203)$$

Substituting this into (199), we obtain

$$\mathbb{E}(U_{[d]}^\top Z_{[d]})^2 = d \cdot \frac{n(n-d)}{(n+2)(n-1)}, \quad (204)$$

as claimed.

Finally, we move on to establish (165). It is easy to see that the expectation of all terms including  $Z_1, Z_2$  is zero, and therefore

$$\mathbb{E}(e_{\mathcal{M}} e_{\bar{\mathcal{M}}}) = \phi(D) \mathbb{E} \left[ \left( \rho \|U_{[d]}\|^2 + \sqrt{1-\rho^2} U_{[d]}^\top Z_{[d]} \right) \left( \rho \|U_{[\bar{d}]}\|^2 + \sqrt{1-\rho^2} U_{[\bar{d}]}^\top Z_{[\bar{d}]} \right) \right], \quad (205)$$

where  $[\bar{d}] = [n] \setminus [d] = \{d+1, \dots, n\}$ . By the same consideration as above, the cross terms are zero. Note that by definition,  $\|U_{[d]}\|^2 + \|U_{[\bar{d}]}\|^2 = \|U\|^2 = n$ , and we therefore have

$$\mathbb{E}(e_{\mathcal{M}} e_{\bar{\mathcal{M}}}) = \phi(D) \left[ \rho^2 \mathbb{E}(\|U_{[d]}\|^2(n - \|U_{[d]}\|^2)) + (1 - \rho^2) \mathbb{E} \left( \sum_{i=1}^d U_i Z_i \sum_{j=d+1}^n U_j Z_j \right) \right] \quad (206)$$

$$= \phi(D) \left[ \rho^2 \mathbb{E}(\|U_{[d]}\|^2(n - \|U_{[d]}\|^2)) + (1 - \rho^2) d(n-d) \mathbb{E}(U_1 U_2 Z_1 Z_2) \right] \quad (207)$$

$$= \phi(D) \left[ \rho^2 \left( dn - \frac{n}{n+2} d(d+2) \right) + (1 - \rho^2) d(n-d) \nu \right]. \quad (208)$$

Recalling that

$$\nu = -\frac{\xi}{n-1} = -\frac{n}{(n+2)(n-1)}, \quad (209)$$

and substituting this into (208), we obtain

$$\mathbb{E}(e_{\mathcal{M}} e_{\bar{\mathcal{M}}}) = \phi(D) \left[ \rho^2 \frac{n}{n+2} d(n-d) - (1 - \rho^2) \frac{n}{n+2} \frac{d(n-d)}{(n-1)} \right] \quad (210)$$

$$= \phi(D) \frac{n}{(n+2)(n-1)} d(n-d) (\rho^2 n - 1). \quad (211)$$

Our claim follows since  $\frac{n}{(n+2)(n-1)} \leq \frac{1}{n}$  for any  $n > 1$ .

## REFERENCES

- [1] V. Strassen, "Gaussian elimination is not optimal," *Numerische mathematik*, vol. 13, no. 4, pp. 354–356, 1969.
- [2] A. Fawzi, M. Balog, A. Huang, T. Hubert, B. Romera-Paredes, M. Barekatain, A. Novikov, F. J. R. Ruiz, J. Schrittwieser, G. Swirszcz *et al.*, "Discovering faster matrix multiplication algorithms with reinforcement learning," *Nature*, vol. 610, no. 7930, pp. 47–53, 2022.
- [3] R. Duan, H. Wu, and R. Zhou, "Faster matrix multiplication via asymmetric hashing," in *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2023, pp. 2129–2138.
- [4] V. V. Williams, Y. Xu, Z. Xu, and R. Zhou, *New Bounds for Matrix Multiplication: from Alpha to Omega*, 2024, pp. 3792–3835.
- [5] A. Tseng, J. Chee, Q. Sun, V. Kuleshov, and C. De Sa, "Quip#: Even better llm quantization with hadamard incoherence and lattice codebooks," *arXiv preprint arXiv:2402.04396*, 2024.
- [6] S. Ashkboos, A. Mohtashami, M. L. Croci, B. Li, M. Jaggi, D. Alistarh, T. Hoefler, and J. Hensman, "Quarot: Outlier-free 4-bit inference in rotated llms," *arXiv preprint arXiv:2404.00456*, 2024.
- [7] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: <http://arxiv.org/abs/1810.04805>
- [8] Y. Polyanskiy and Y. Wu, *Information theory: From coding to learning*. Cambridge university press, 2024.
- [9] O. Ordentlich and U. Erez, "A simple proof for the existence of good pairs of nested lattices," *IEEE Transactions on Information Theory*, vol. 62, no. 8, pp. 4439–4453, 2016.
- [10] M. W. Mahoney *et al.*, "Randomized algorithms for matrices and data," *Foundations and Trends® in Machine Learning*, vol. 3, no. 2, pp. 123–224, 2011.
- [11] P.-G. Martinsson and J. A. Tropp, "Randomized numerical linear algebra: Foundations and algorithms," *Acta Numerica*, vol. 29, pp. 403–572, 2020.

- [12] P. Drineas, R. Kannan, and M. W. Mahoney, “Fast monte carlo algorithms for matrices i: Approximating matrix multiplication,” *SIAM Journal on Computing*, vol. 36, no. 1, pp. 132–157, 2006.
- [13] M. S. Charikar, “Similarity estimation techniques from rounding algorithms,” in *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*, 2002, pp. 380–388.
- [14] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, “Locality-sensitive hashing scheme based on p-stable distributions,” in *Proceedings of the twentieth annual symposium on Computational geometry*, 2004, pp. 253–262.
- [15] R. Pagh, “Compressed matrix multiplication,” *ACM Transactions on Computation Theory (TOCT)*, vol. 5, no. 3, pp. 1–17, 2013.
- [16] Y. Guo, A. Yao, and Y. Chen, “Dynamic network surgery for efficient dnns,” *Advances in neural information processing systems*, vol. 29, 2016.
- [17] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, “Quantized neural networks: Training neural networks with low precision weights and activations,” *Journal of Machine Learning Research*, vol. 18, no. 187, pp. 1–30, 2018.
- [18] Y. Gong, L. Liu, M. Yang, and L. Bourdev, “Compressing deep convolutional networks using vector quantization,” *arXiv preprint arXiv:1412.6115*, 2014.
- [19] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. 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.
- [20] T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer, “Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 30 318–30 332, 2022.
- [21] Z. Yao, R. Yazdani Aminabadi, M. Zhang, X. Wu, C. Li, and Y. He, “Zeroquant: Efficient and affordable post-training quantization for large-scale transformers,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 168–27 183, 2022.
- [22] G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han, “Smoothquant: Accurate and efficient post-training quantization for large language models,” in *International Conference on Machine Learning*. PMLR, 2023, pp. 38 087–38 099.
- [23] S. Ma, H. Wang, L. Ma, L. Wang, W. Wang, S. Huang, L. Dong, R. Wang, J. Xue, and F. Wei, “The era of 1-bit llms: All large language models are in 1.58 bits,” *arXiv preprint arXiv:2402.17764*, 2024.
- [24] E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh, “OPTQ: Accurate quantization for generative pre-trained transformers,” in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: <https://openreview.net/forum?id=tcbBPnfwxS>
- [25] H. Jegou, M. Douze, and C. Schmid, “Product quantization for nearest neighbor search,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 117–128, 2010.
- [26] D. Blalock and J. Gutttag, “Multiplying matrices without multiplying,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 992–1004.
- [27] P. Stock, A. Joulin, R. Gribonval, B. Graham, and H. Jégou, “And the bit goes down: Revisiting the quantization of neural networks,” *arXiv preprint arXiv:1907.05686*, 2019.
- [28] R. Guo, S. Kumar, K. Choromanski, and D. Simcha, “Quantization based fast inner product search,” in *Artificial intelligence and statistics*. PMLR, 2016, pp. 482–490.
- [29] D. Malak, “Distributed structured matrix multiplication,” *arXiv preprint arXiv:2405.02904*, 2024.
- [30] D. Krithivasan and S. S. Pradhan, “Lattices for distributed source coding: Jointly gaussian sources and reconstruction of a linear function,” *IEEE Transactions on Information Theory*, vol. 55, no. 12, pp. 5628–5651, 2009.
- [31] A. B. Wagner, “On distributed compression of linear functions,” *IEEE Transactions on Information Theory*, vol. 57, no. 1, pp. 79–94, 2010.
- [32] A. Ingber, T. Courtade, and T. Weissman, “Compression for quadratic similarity queries,” *IEEE transactions on information theory*, vol. 61, no. 5, pp. 2729–2747, 2015.
- [33] I. Ochoa, A. Ingber, and T. Weissman, “Compression schemes for similarity queries,” in *2014 Data Compression Conference*, 2014, pp. 332–341.
- [34] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, “Straggler mitigation in distributed matrix multiplication: Fundamental limits and optimal coding,” *IEEE Transactions on Information Theory*, vol. 66, no. 3, pp. 1920–1933, 2020.
- [35] A. Adler, J. Tang, and Y. Polyanskiy, “Efficient representation of large-alphabet probability distributions,” *IEEE Journal on Selected Areas in Information Theory*, vol. 3, no. 4, pp. 651–663, 2022.
- [36] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Springer Science & Business Media, 2012, vol. 159.
- [37] R. Zamir, *Lattice Coding for Signals and Networks: A Structured Coding Approach to Quantization, Modulation, and Multiuser Information Theory*. Cambridge University Press, 2014.
- [38] U. Erez and R. Zamir, “Achieving  $1/2 \log(1 + \text{snr})$  on the AWGN channel with lattice encoding and decoding,” *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2293–2314, 2004.
- [39] A. J. Stam, “Limit theorems for uniform distributions on spheres in high-dimensional euclidean spaces,” *Journal of Applied probability*, vol. 19, no. 1, pp. 221–228, 1982.
- [40] O. Ordentlich, O. Regev, and B. Weiss, “Bounds on the density of smooth lattice coverings,” *arXiv preprint arXiv:2311.04644*, 2023.
- [41] G. Forney and G. Ungerboeck, “Modulation and coding for linear gaussian channels,” *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2384–2415, 1998.
- [42] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*. New York: Springer-Verlag, 1988.
- [43] E. Agrell and B. Allen, “On the best lattice quantizers,” *IEEE Transactions on Information Theory*, vol. 69, no. 12, pp. 7650–7658, 2023.
- [44] M. S. Viazovska, “The sphere packing problem in dimension 8,” *Annals of mathematics*, pp. 991–1015, 2017.
- [45] J. Conway and N. Sloane, “Fast quantizing and decoding and algorithms for lattice quantizers and codes,” *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 227–232, 1982.
- [46] H. Cohn and A. Kumar, “Optimality and uniqueness of the leech lattice among lattices,” *Annals of mathematics*, pp. 1003–1050, 2009.
- [47] H. Cohn, A. Kumar, S. Miller, D. Radchenko, and M. Viazovska, “The sphere packing problem in dimension 24,” *Annals of mathematics*, vol. 185, no. 3, pp. 1017–1033, 2017.
- [48] A. Vardy and Y. Be’ery, “Maximum likelihood decoding of the leech lattice,” *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1435–1444, 1993.
- [49] A. Vardy, “Even more efficient bounded-distance decoding of the hexacode, the golay code, and the leech lattice,” *IEEE Transactions on Information Theory*, vol. 41, no. 5, pp. 1495–1499, 1995.
- [50] E. Viterbo and J. Boutros, “A universal lattice code decoder for fading channels,” *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1639–1642, 1999.



- [51] J. Conway and N. Sloane, "Voronoi regions of lattices, second moments of polytopes, and quantization," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 211–226, 1982.
- [52] N. Ailon and B. Chazelle, "The fast johnson–lindenstrauss transform and approximate nearest neighbors," *SIAM Journal on computing*, vol. 39, no. 1, pp. 302–322, 2009.
- [53] E. Liberty, *Accelerated dense random projections*. Yale University New Haven, CT, 2009.
- [54] J. A. Tropp, "Improved analysis of the subsampled randomized hadamard transform," *Advances in Adaptive Data Analysis*, vol. 3, no. 01n02, pp. 115–126, 2011.