High-Rate Quantized Matrix Multiplication II

Or Ordentlich and Yury Polyanskiy

***Abstract*—This is the second part of the work investigating quantized matrix multiplication (MatMul). In part I we considered the case of calibration-free quantization, whereas here we discuss the setting where covariance matrix Σ_X of the columns of the second factor is available. This setting arises in the ubiquitous task of *weight-only* post-training quantization of LLMs.**

Weight-only quantization is related to the problem of weighted mean squared error (WMSE) source coding, whose classical (reverse) waterfilling solution dictates how one should distribute rate between coordinates of the vector. We show how waterfilling can be used to improve practical LLM quantization algorithms (GPTQ), which at present allocate rate equally. A recent scheme (known as "WaterSIC") that only uses scalar INT quantizers is analyzed and its high-rate performance is shown to be (a) basis free (i.e., characterized by the determinant of Σ_X and, thus, unlike existing schemes, is immune to applying random rotations); and (b) within a multiplicative factor of $\frac{2\pi e}{12}$ (or 0.25 bit/entry) of the information-theoretic distortion limit. GPTQ's performance, in turn, is affected by the choice of basis, but for a random rotation and actual Σ_X from Llama-3-8B we find it to be within 0.1 bit (depending on the layer type) of WaterSIC, suggesting that GPTQ with random rotation is also near optimal, at least in the high-rate regime.

I. Introduction

Matrix multiplication (MatMul) constitutes the main building block of AI, and quantization for matrix multiplication has become a central technique for increasing its efficiency. In Part I [1] we discussed the problem of quantizing two generic matrices with the goal of approximating their product with minimal distortion. The fundamental limits for this assumptions-free setup were derived in [2], which we compared with the performance of popular generic quantization schemes for MatMul (such as ubiquitous absmax round-to-nearest with INT or FP arithmetic).

Generic quantized MatMul schemes require no prior assumptions on the matrices to be multiplied, and their performance is consequently robust. However, in LLMs activations often have a rather special low-dimensional structure, obviating the need for orthogonalizing quantization errors with the principal directions of activation variation [3], [4]. The process of collecting statistics of activation is known as *calibration*, and thus we can think of Part I as focusing on calibration-free and Part on calibration-aided methods. Clearly, the goal of such methods is to improve matrix product approximation by leveraging calibration statistics.

Furthermore, in this paper we focus on the case where arithmetic (actual matmul) is executed and activations loaded in full precision. The goal, thus, is to load weight matrices in compressed (low-resolution) format from the high bandwidth memory (HBM) and then dequantize them into full-precision matrix before executing the actual matmul. Compared to full-precision (BF16) weights, quantizing those to R bits per parameter reduces the required HBM storage and bandwidth capacity by factor $16/R$.

In Section II we study the fundamental limits of *weight-only quantization* where we are interested in the matrix product $X^\top W$ and only W needs to be quantized to $\hat{W}$, whereas X is given in full resolution. The problem is made interesting through the fact that while quantization of W cannot know X, it has access to covariance matrix Σ_X of (the columns of) X. We show that, in essence, using standard "isotropic" codebook and Σ_X-oblivious quantization, one attains distortion $D \approx (\frac{1}{n}\operatorname{tr}\Sigma_X)2^{-2R}$. At the same time, the classical waterfilling formula says that optimal codebook (with optimal quantization) can attain $D \approx |\Sigma_X|^{1/n}2^{-2R}$, with the difference given by the gap in AM-GM inequality for eigenvalues of Σ_X. It turns out that even with isotropic codebook, but Σ_X-aware encoding one can still attain the same performance.

In Section III we discuss the practical aspects of the weights-only quantization problem. Low-complexity algorithms for weight-only quantization in LLMs have been pioneered by [5] (a Σ_X-oblivious rounding to nearest integer), [3], and GPTQ [4]. The latter was originally derived from [6], but was soon found to be equivalent to classical algorithms known as *successive interference cancellation (SIC)* in communication and Babai's algorithm in computer science, cf. Section III. We show that GPTQ (that uses constant rate per entry) attains distortion decaying as $D \approx \frac{2\pi e}{12}(\frac{1}{n}\sum_{i=1}^n U_{i,i}^2)2^{-2R}$, where $U^\top U = \Sigma_X$ is the upper-triangular (Cholesky) decomposition of Σ_X. At the same time, by adjusting rate per-coordinate ("WaterSIC" algorithm [7]) one can attain $D \approx \frac{2\pi e}{12}(\prod_{i=1}^n U_{i,i}^2)^{\frac{1}{n}}2^{-2R}$. Since $\prod_{i=1}^n U_{i,i} =$

O. Ordentlich is with the Hebrew University of Jerusalem, Israel (or.ordentlich@mail.huji.ac.il). Y. Polyanskiy is with the MIT, USA (yp@mit.edu). The work of OO was supported by the Israel Science Foundation (ISF), grant No. 2878/25. The work of YP was supported in part by the MIT-IBM Watson AI Lab and by the National Science Foundation under Grant No CCF-2131115.

$\sqrt{|\Sigma_X|}$ this performance is only a factor away from information-theoretically optimal, or within a rate gap of at most 0.25 bit.

Figures 1 and 5 exemplify rate gains for all these four (two theoretical and two practical) algorithms on the example of layers of Llama-3-8B and wiki2 calibration data.

II. Weight quantization: Theory

Let us restrict attention to a particular linear layer in the network with weight matrix $W \in \mathbb{R}^{n\times a}$. The linear layer operates by taking (a vector of) input activations X and producing

$$Y = X^\top W\,.$$

For that, W needs to be loaded from memory and this results in the crucial bottleneck during generation/reasoning part of the LLM operation. Thus, our job is to replace W with quantized version $\hat{W}$, which can be loaded much faster, with the target of minimizing *expected* distortion over random inputs X, that is we aim to minimize

$$\begin{aligned} D &= \frac{1}{n}\mathbb{E}\|X^\top(W-\hat{W})\|_F^2 \\ &= \frac{1}{n}\operatorname{tr}(W-\hat{W})^\top\Sigma_X(W-\hat{W})\,, \end{aligned} \tag{1}$$

where $\Sigma_X = \mathbb{E}[XX^\top]$ (This objective is by far the most popular one, though, others exist [8], [9], [10].)

Since only the second order statistics Σ_X affect this objective, in practice one uses a set of samples (known as *calibration data*) from the underlying distribution in order to estimate it. With the second order statistics at hand, the problem of quantizing W becomes a weighted mean squared error (WMSE) quantization problem. We note that calibration data about X is also useful for quantization of both weights and activations (i.e. the setting of [1]). See e.g. [11], [12, Section 4.5], [13], [14].

Weight matrices start life as iid Gaussian and evolve during training. However, in many ways their statistics are still largely very similar to iid Gaussian [7, Appendix] and that is how we will model them in this (theoretical) section. Some (mostly older) LLMs have peculiar aspects of weight matrices, such as existence of outliers and rank deficiencies, that need to be exploited/mitigated in practical algorithms, though.

Notice also that objective (1) treats distortion across columns of W in an additive manner. Consequently, analyzing the case of $a = 1$ (single column, or single output neuron) can be done without loss of generality, at least as long as $n \gg 1$ and full advantage of vector quantization can be exploited already in dimension n, without needing to do joint vector quantization across all na dimensions.

So in summary, the goal of this section is to understand theoretically the problem of mapping $W \sim \mathcal{N}(0, I_n)$ to $\hat{W}$ belonging to the universe of 2^{nR} possibilities, with the goal of minimizing

$$D = \frac{1}{n}(W-\hat{W})^\top\Sigma_X(W-\hat{W})\,. \tag{2}$$

One important aspect of the weight only quantization problem worth pointing out from the outset is the asymmetry between encoding and decoding. The encoding (quantization) is done offline, once for each model, and may therefore be computationally demanding. The decoding (de-quantization), on the other hand, is executed online every time a weight matrix is used, and must therefore be highly efficient. (This is in contrast with activation quantization, which is done online and has to be extremely computationally efficient.) In particular, while information about Σ_X is available to the encoder (recall that it operates offline and can be quite slow), the decoder's online operation requires all information about Σ_X be packaged inside the rate-constrained description of W. Thus, assuming decoder knows Σ_X or its SVD basis (rather than only its spectrum) is not reflecting the practical constraints adequately.

Nevertheless, below we will start by treating the impractical *fully informed case*, which will provide a firm lower bound to the more practical version of *uninformed/oblivious decoder case*. We initiate our discussion with a simple heuristic demonstration of the main points on the example of scalar quantization.

A. Scalar quantization and waterfilling

Before discussing information theoretic results, let us consider a very simple case of X with uncorrelated coordinates, i.e. $\Sigma_X = \operatorname{diag}\{\lambda_1,\dots,\lambda_n\}$. Suppose, in addition, that the weights satisfy $W_i \in \operatorname{Uniform}[-1/2, 1/2)$.

Consider, first, the case of usual uniform quantization using ϵ-grid, where $\epsilon = 2^{-R}$. The resulting distortion (in accordance with additive noise approximation discussed in Part I [1]) will be

$$D_1(R) = \frac{1}{n}\sum_{i=1}^n \lambda_i\frac{\epsilon^2}{12} = \frac{\bar{\lambda}_A}{12}2^{-2R}\,,$$

where $\bar{\lambda}_A = \frac{1}{n}\sum_i \lambda_i$ is the arithmetic mean of λ's.

Now, given that different coordinates of W contribute differently to D, one may naturally try to allocate rate more judiciously. Specifically, let us solve the problem of

$$D_2 = \min \frac{1}{12n}\sum_{i=1}^n \lambda_i\epsilon_i^2\,,$$

subject to rate constraint $\prod \frac{1}{\epsilon_i} \le 2^{nR}$. Using Lagrange multipliers we can easily find a parametric formula for the optimizer, given in terms of parameter $\tau > 0$:

$$D_2 = \frac{1}{n}\sum_{i=1}^{n} \min(\tau, \lambda_i/12)$$
$$R = \frac{1}{2n}\sum_{i=1}^{n} \log \max(1, \frac{\lambda_i}{12\tau}) ,$$

where the grid spacings $\epsilon_i = \min(\sqrt{\frac{12\tau}{\lambda_i}}, 1)$. This kind of allocation is traditionally called (reverse) *waterfilling solution*, due to interpretation of τ as water level that reduces i'th coordinate baseline distortion (equal to $\lambda_i/12$) down to τ. Those coordinates that are so insignificant ("below waterlevel") get zero rate allocation and are quantized to zero.

In accordance with our focus on high-rate case, we can assume that $\tau < \min_i \lambda_i$ and get a simplified expression:

$$D_2(R) = \frac{\bar{\lambda}_G}{12} 2^{-2R} ,$$

where $\bar{\lambda}_G = (\prod_i \lambda_i)^{1/n}$ is the geometric mean of λ_i's.

In all, we conclude that when quantizing vectors under weighted quadratic metric, one should choose quantization grid spacing $\epsilon_i \propto \frac{1}{\sqrt{\lambda_i}}$ with coefficient of proportionality chosen depending on the target required rate R. How much does one win from this optimization is given by the AM-GM inequality $\bar{\lambda}_G < \bar{\lambda}_A$. Another way to put this is that optimizing per-coordinate quantizers one wins about $\frac{1}{2}\log \frac{\bar{\lambda}_A}{\bar{\lambda}_G}$ bits of rate.

We will return to this idea below, when we describe a practical WaterSIC algorithm for weight-only quantization in Section III-D. There the role of λ_i's is played by the diagonal elements of the Cholesky decomposition of (non-diagonal) Σ_X.

Fig. 1 illustrates rate-advantage for input activations to various linear layers of Llama-3-8B.

B. WMSE Quantization: information theoretic limit

The waterfilling solution derived heuristically in the previous section can be in fact made rigorous, at least for the case of Gaussian weights $W \sim \mathcal{N}(0, \sigma_W^2 I_n)$, which we consider here. Another assumption we make is that the covariance matrix $\Sigma_X \in \mathcal{PSD}_n$ (the set of all $n \times n$ PSD matrices) is fixed and known to both encoder and decoder.

The formal problem is this: given $W \sim \mathcal{N}(0, \sigma_W^2 I_n)$, encoder $f : \mathbb{R}^n \to [2^{nR}]$ produces a rate-R description of W. Subsequently, decoder $g : [2^{nR}] \to \mathbb{R}^n$ converts this description into the estimate $\hat{W}$. The information-theoretic question is to determine the value of the smallest distortion attainable by the best pair (f, g), i.e.

$$D^*(\Sigma_X, R) = \min_{f,g} \frac{1}{n}\mathbb{E}[(W - \hat{W})^\top \Sigma_X (W - \hat{W})] .$$

The naive (suboptimal) solution is to use a standard isotropic Gaussian codebook of rate R, which results in error covariance given by $\mathbb{E}[(W - \hat{W})(W - \hat{W})^\top] = \sigma_W^2 2^{-2R} I_n$. Note that the previous result is simple if we understand $\mathbb{E}$ as averaging over the Gaussian codebook, but is rather non-trivial if we want to demonstrate existence of a codebook with nearly-white covariance matrix of errors. This was the main technical difficulty behind the results of [2], cf. Theorem 13 there. Nevertheless, under this assumption on the error distribution, we get

$$D_{iso}(R) = 2^{-2R} \frac{\sigma_W^2}{n} \operatorname{tr} \Sigma_X . \tag{3}$$

Although this is generally very far from $D^*(\Sigma_X, R)$, such scheme works simultaneously for all Σ_X and does not require knowledge thereof.

Scheme that is allowed to fully exploit knowledge of Σ_X can first compute $W' = V^\top W$, where V is the orthogonal matrix in the SVD decomposition of $\Sigma_X = V^\top \Lambda V$. If decoder can estimate $\hat{W}'$, then it can also set $\hat{W} = V\hat{W}'$. Then, the distortion can be expressed in terms of $(W', \hat{W}')$ as

$$D = \frac{1}{n}\sum_{i=1}^{n} \lambda_i \mathbb{E}[(\hat{W}'_i - W'_i)^2] .$$

We see that after this change of coordinates, the problem indeed becomes that of *weighted* mean-squared error (WMSE), justifying the name.

To relate distortion and rate, we consider a standard data-processing argument, see [15, Section 23.4]:

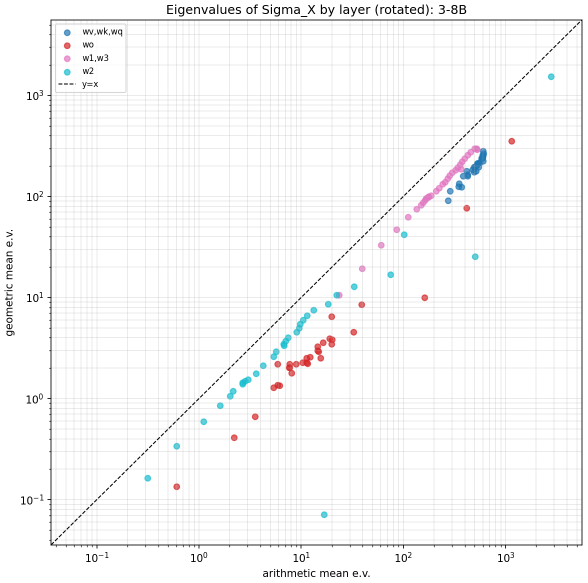
$$nR \ge I(W'; \hat{W}') \ge \sum_{i=1}^{n} I(W'_i; \hat{W}'_i) ,$$

where the second inequality is due to independence of coordinates of $W' \sim \mathcal{N}(0, \sigma_W^2 I_n)$, cf. [15, Theorem 6.1]. Now, given value $D_i = \mathbb{E}[(\hat{W}'_i - W'_i)^2]$ the smallest $I(W'_i; \hat{W}'_i) = \frac{1}{2}\log \frac{\sigma_W^2}{D_i}$ is attained under Gaussian coupling, cf. [15, Section 26.1.2]. Overall, we get

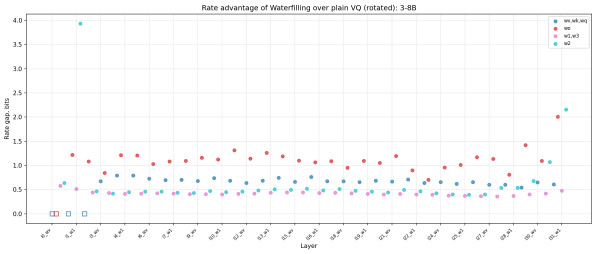
$$R \ge \frac{1}{2n}\sum_{i=1}^{n} \log \frac{\sigma_W^2}{D_i}$$
$$D = \frac{1}{n}\sum_{i=1}^{n} \lambda_i D_i .$$

Minimizing the value of D given R can be done via Lagrange multipliers, resulting in the (reverse) *waterfilling solution* parameterized by $\tau > 0$:

$$D^*(\tau) = \frac{\sigma_W^2}{n}\sum_{i=1}^{n} \min\{\lambda_i, \tau\},$$
$$R^*(\tau) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}\log \max\left\{1, \frac{\lambda_i}{\tau}\right\} . \tag{4}$$



(a) Scatter plot illustrating gap in AM-GM inequality for eigenvalues of Σ_X.



(b) Rate advantage (in high-rate regime) of waterfilling over D_{iso} (which corresponds to the case of Σ_X-oblivious encoder and decoder). Squares correspond to (skipped) layers with singular Σ_X.

Fig. 1: Illustrating Σ_X of activations entering various layers of Llama-3-8B when processing Wikitext-2 dataset. Note that this is an estimate of the rate advantage assumes weight matrices are well modeled by $\mathcal{N}(0, I_n)$. In particular, actual weight matrices were never used for this plot.

Note that, while our argument only gives a lower bound on the minimal distortion, it can be shown to be achievable asymptotically for large n, even in the single vector setting [16, Proposition 2.1]. Thus, in the remainder of this paper, we will consider the value $D^*(R)$ given by waterfilling to be the actual fundamental limit, ignoring small non-asymptotic penalty.

Just like in the previous heuristic section, we see that optimal codebook allocates rate unequally: the PCA direction corresponding to high values of λ_i should get quantized more finely with $D_i \propto \frac{1}{\lambda_i}$. In the high-rate regime we get

$$D^*_{\mathrm{High-Rate}}(R) = |\Sigma_X|^{1/n}\sigma_W^2 2^{-2R}\,, \tag{5}$$

and the advantage compared to uninformed isotropic coding is again given by the gap in the AM-GM inequality, or

$$\frac{1}{2}\log\frac{\frac{1}{n}\operatorname{tr}\Sigma_X}{|\det\Sigma_X|^{1/n}}$$

bits in effective rate.

C. Σ_X-oblivious decoder

In the setup above, we have allowed the decoder to depend on Σ_X (in particular, to apply SVD decomposition). However, in LLM applications this would require communicating Σ_X (or the SVD basis matrix V) to GPU's compute devices, which would be very costly. Instead, we would like to use a very lean decoder, that only uses those bits provided with the description of W to produce its reconstruction, universally for all Σ_X.

We will therefore require that while the encoder $f(W, \Sigma_X)$ may depend on Σ_X, the decoder $g : [2^{nR}] \to \mathbb{R}^n$ depends only on the bits it receives from the encoder. With this constraint, the quantization problem consists of a codebook $\mathcal{C} \subset \mathbb{R}^n$ with $|\mathcal{C}| \le 2^{nR}$ codewords, agreed upon by the encoder and decoder, where the encoder's job is to solve, or approximately solve, the problem

$$\hat{W}^* = \operatorname*{argmin}_{c\in\mathcal{C}}[(W-c)^\top\Sigma_X(W-c)]. \tag{6}$$

The encoder then describes $\hat{W}^*$ using nR bits, and the decoder can reconstruct $\hat{W}^*$ from those bits.

We know from the previous section that in order to achieve the informed (waterfilling) $D^*(R)$ optimal codebook needs to be anisotropic: the "grids" need to be denser along the PCA directions corresponding to higher values of λ_i. In the uninformed case, we are not able to do such adaptation (since Σ_X is unknown at the time of the codebook design). Thus, we are forced to use isotropic codebook. In [16] a somewhat surprising result is shown: if one generates $\mathcal{C}$ iid from isotropic Gaussian distribution, then simultaneously for all Σ_X the adaptive encoder (6) achieves the following (parametric)

rate-distortion:

$$D_{rc}(R) = \frac{\sigma_W^2}{n}\sum_{i=1}^{n}\frac{\lambda_i}{1+\lambda_i T}$$
$$R = \frac{1}{2n}\sum_{i=1}^{n}\log(1+\lambda_i T)\,.$$

First, observe that in the high-rate regime $R \gg 1$ (i.e. $T \gg 1$), we get

$$D_{rc}(R) = |\Sigma_X|^{1/n}\sigma_W^2 2^{-2R} + O(2^{-4R})\,.$$

Comparing this to (5), we observe that in the high-rate regime this scheme attains the optimal waterfilling distortion (upto higher order corrections), *despite decoder not knowing* Σ_X. Furthermore, analysis in [16] shows that universally over all possible Σ_X the gap between waterfilling solution $D^*(R)$ and $D_{rc}(R)$ is never more than 0.11 bit.

This result demonstrates that the most significant part of the gap between (3) and (4) is not due to suboptimal codebook design, but rather due to suboptimal rounding to that codebook. In other words, one must focus on implementing better rounding schemes rather than optimizing codebooks. Unfortunately, as we will see in the next section, even for the uniform quantization ($\mathbb{Z}^n$ codebook), solving (6) is computationally hard. Thus, despite theoretical near-optimality of isotropic codebooks, the algorithmic difficulty makes Σ_X-oblivious quantization very interesting and rather underexplored. A practical low-complexity algorithm, known variously as successive-interference cancellation (SIC), Babai's algorithm, GPTQ or LDLQ, will be discussed in the next section.

III. Weight quantization: Practice

In this section we discuss approximate solutions for the WMSE quantization problem, that can be computed efficiently. We will describe several variants of the GPTQ/LDLQ algorithms [4], [17], and propose novel improvements based on our information theoretic analysis. The schemes we discuss rely on a reformulation of the "optimal rounding" problem (6) using the Cholesky decomposition [18], [19]. We develop this reformulation in Subsection III-A. The analysis of the quantization schemes that follow requires some background in high-resolution quantization using lattices. We provide this background in Subsection III-B, and then in Subsection III-C we explain and analyze the SIC rounding algorithm. In its most basic form, this algorithm is equivalent to the canonical GPTQ/LDLQ [4], [17], as was recently observed in [18], [19]. In light of the information theoretic analysis above, and the analysis of the SIC algorithm, it immediately becomes clear that a simple variation of GPTQ/LDLQ called *WaterSIC*, that we develop in Subsection III-D, provides an improved distortion and is provably quite close to the lower bound (5) we developed on the WMSE. The WaterSIC algorithm is further developed in [7], and is also shown to perform remarkably well for low-rate weight only quantization of LLMs.

A. Reformulation of (6) via Cholesky Decomposition

Using the Cholesky decomposition, we can decompose Σ_X as $\Sigma_X = U^\top U$ where $U \in \mathbb{R}^{n\times n}$ is an upper triangular matrix. Plugging this into (6) we obtain that given a fixed codebook $\mathcal{C}$ the optimal quantization problem becomes

$$\hat{W}^* = \operatorname*{argmin}_{c\in\mathcal{C}} \|Y - U\cdot c\|^2, \quad \text{where } Y = UW \in \mathbb{R}^n. \tag{7}$$

Denote

$$e_Y = Y - U\hat{W}^*, \quad e_W = W - \hat{W}^*, \tag{8}$$

and note that

$$\hat{W}^* = W + e_W = W + U^{-1}e_Y. \tag{9}$$

Recall that since $W \sim \mathcal{N}(0,\sigma_W^2 I_n)$, we have that $\Pr(W \notin \sqrt{n(1+\varepsilon)\sigma_W^2}\mathcal{B}) \to 0$ for any $\varepsilon > 0$, as $n\to\infty$, where $\mathcal{B} = \{x \in \mathbb{R}^n : \|x\| \le 1\}$. For high-resolution quantization, where e_Y is so small that $U^{-1}e_Y$ can also be assumed small, this will imply that $\hat{W}^*$ will typically also fall inside a ball with radius $\sqrt{n(1+\varepsilon_R)\sigma_W^2}\mathcal{B}$, where ε_R vanishes with R (and n). This fact will be useful in the analysis below.

Before proceeding further, we note that the diagonal entries of the Cholesky matrix U will play here the same role as eigenvalues of Σ_X did in Section II. In particular, we will see that algorithms whose codebooks have equal density in each direction (SIC/GPTQ) attain error

$$D_{\mathrm{GPTQ}}(R) \approx \frac{2\pi e}{12}\sigma_W^2\left(\frac{1}{n}\sum_i U_{i,i}^2\right)2^{-2R}, \tag{10}$$

which depends on *arithmetic* mean of squared entries of diagonal of U (which is highly dependent on permutation of columns of Σ_X and application of random rotations). Our new scheme ("WaterSIC") spends rate more judiciously (more bits to coordinates with higher $U_{i,i}$ and attains

$$D_{\mathrm{WaterSIC}}(R) \approx \frac{2\pi e}{12}\sigma_W^2\left(\prod_i U_{i,i}^2\right)^{1/n}2^{-2R}. \tag{11}$$

On a first sight, we only get the same AM-GM improvement as with / without waterfilling. However, what is remarkable and much deeper is the fact that since $|\Sigma_X| = |U|^2$, we have $(\prod_i U_{i,i}^2)^{1/n} = |\Sigma_X|^{1/n}$, thus recovering (within factor $\frac{2\pi e}{12}$, which corresponds to Koshelev's famous 0.254 bit gap [15, Section 24.1.5]) the information-theoretic optimal waterfilling rate $D^*_{\mathrm{High-Rate}}$, see (5).

B. Preliminaries on High-resolution quantization via shaped/entropy coded lattice quantizers

Lattices: We review some basic lattice definitions. See [20] for a comprehensive treatment of lattices in information theory. Any lattice $L \subset \mathbb{R}^n$ has a (non-unique) generating matrix $G \in \mathbb{R}^{n\times n}$ such that $L = G\mathbb{Z}^n$. The covolume of the lattice L, denoted $\mathrm{covol}(L)$) is defined as $|G|$. The point density of a lattice is $\gamma(L) = \mathrm{covol}^{-1}(L) = |G|^{-1}$. For a lattice $L \subset \mathbb{R}^n$ we define the nearest neighbor quantizer $Q_L : \mathbb{R}^n \to L$ as

$$Q_L(x) = \underset{\lambda\in L}{\mathrm{argmin}} \|x - \lambda\|, \tag{12}$$

where ties are broken arbitrarily, but in systematic manner. The Voronoi region $\mathcal{V}_L$ is defined as the set of all points in $\mathbb{R}^n$ that are closer to 0 than to any other lattice point

$$\mathcal{V}_L = \{x \in \mathbb{R}^n \ : \ Q_L(x) = 0\}. \tag{13}$$

The volume of the Voronoi region (as well as any other fundamental cell of L) is $\mathrm{Vol}(\mathcal{V}_L) = \mathrm{covol}(L) = |G|$.

We define the second moment of the lattice L as

$$\sigma^2(L) = \frac{1}{n}\mathbb{E}\|Z\|^2, \tag{14}$$

where $Z \sim \mathrm{Uniform}(\mathcal{V}_L)$ is a random vector uniformly distributed over the Voronoi region of L.

High-resolution lattice quantizers: Designing a quantizer for the WMSE problem with Σ_X-oblivious decoder entails choosing the codebook $\mathcal{C}$, which consists of 2^{nR} vectors in $\mathbb{R}^n$. One way to construct such a codebook is to start with an infinite constellation, in our case a lattice $L \subset \mathbb{R}^n$, and then choose from L only 2^{nR} points [20]. This can be done by choosing a shaping region $\mathcal{S} \subset \mathbb{R}^n$, e.g. $\mathcal{S} = r\mathcal{B}$ for some $r > 0$, and taking $\mathcal{C} = L\cap\mathcal{S}$ where $\mathcal{S}$ is dilated such that $|L \cap \mathcal{S}| = 2^{nR}$. Correspondingly, this way of thinking about vector quantization splits the problem into design of good infinite constellations (with prescribed point density) and shaping regions that chop off a finite part of it.

Many practical solutions for weight-only quantization that were considered in the literature fall under the shaped lattice quantization paradigm: GPTQ with INT constellations [4] as well as LDLQ with E_8-based codebooks as in QuIP# [17], [21], and many more.

Assume the point $\hat{W}^* \in L$ that minimizes (7) with respect to the entire lattice L (rather than $L \cap \mathcal{S}$) is within the shaping region $\mathcal{S}$. In this case, $e_Y \in \mathcal{V}_{UL}$ and if we further assume $e_Y \sim \mathrm{Uniform}(\mathcal{V}_{UL})$ the distortion is $\sigma^2(UL)$. However, whenever the *overload* event $\hat{W}^* \notin \mathcal{S}$ occurs, the squared error may significantly exceed $\sigma^2(UL)$. In high-resolution quantization, we may assume $\hat{W}^* \approx W$, and the overload probability is well approximated by $\Pr(W \notin \mathcal{S})$. As mentioned above, taking $\mathcal{S} = \sqrt{n(1+\varepsilon)\sigma_W^2}\mathcal{B}$, with some small $\varepsilon > 0$ (or any other $\mathcal{S}$ that contains this ball) suffices for achieving vanishing overload probability. Thus, for such choice of $\mathcal{S}$ we have that $D \approx \sigma^2(UL)$.

If L is "sufficiently dense" with respect to $\mathcal{S}$, it holds that $|L \cap \mathcal{S}| \approx \gamma(L) \cdot \mathrm{Vol}(S)$ where $\gamma(L)$ is the point density of L (see [22, Lemma 3.3] for precise bounds on $|L\cap\mathcal{S}|$). This approximation becomes accurate in the high-resolution limit (as high-resolution corresponds to large $\gamma(L)$). We can therefore approximate the rate as

$$R = \frac{1}{n}\log|L \cap \mathcal{S}| \approx \frac{1}{n}\log(\mathrm{Vol}(\mathcal{S})) + \frac{1}{n}\log\gamma(L). \tag{15}$$

It follows that in the high-resolution regime the quantization rate R and the normalized log point-density $\frac{1}{n}\log\gamma(L)$ are equal up to a constant. Furthermore, recalling that $D \approx \sigma^2(UL)$ provided that $\Pr(W \notin \mathcal{S})$ is sufficiently small, we can express D in the form familiar to us. Specifically, we get from (15)

$$\begin{aligned} D &\approx \left(\sigma^2(UL)\cdot\gamma^{\frac{2}{n}}(UL)\right)|U|^{\frac{2}{n}}\cdot\mathrm{Vol}^{\frac{2}{n}}(\mathcal{S})\cdot 2^{-2R} \\ &= \left(\sigma^2(UL)\cdot\gamma^{\frac{2}{n}}(UL)\right)|\Sigma_X|^{\frac{1}{n}}\cdot\mathrm{Vol}^{\frac{2}{n}}(\mathcal{S})\cdot 2^{-2R}, \end{aligned} \tag{16}$$

where we also used the fact $\gamma(UL) = |U|^{-1}\gamma(L)$. The term in $(\cdot)$ is a famous and extremely well-studied quantity [20] known as the normalized second moment (NSM) of UL, defined as $\sigma^2(UL)\cdot\gamma^{\frac{2}{n}}(UL)$. From (16) we see that the tradeoff between rate and distortion is determined by: 1)The volume of the shaping region (which is required to satisfy $\Pr(W \notin \mathcal{S}) \ll 1$); 2)The NSM of UL.

We stress that $\sigma^2(UL)$ is the WMSE distortion when the $\mathrm{argmin}_{c\in L}$ in (7) is solved exactly. As we discuss below, finding the exact solution to this problem is generally infeasible for large n, and one must resort to approximate solutions. In the case where a sub-optimal quantizer $q : \mathbb{R}^n \to L$ is used, the NSM is replaced with $\frac{1}{n}\mathbb{E}\|Y - U\cdot q(Y)\|^2\cdot\gamma^{\frac{2}{n}}(UL)$ in (16).

An alternative approach for shaping is entropy coding (EC). If quantization with variable rate is allowed, and only the *expected* quantization rate is constrained to be at most R, one can quantize the source to the point $\hat{W}^* \in L$ that minimizes (7) (assuming $\mathcal{C} = L$), and then describe the resulting point in bits using EC.

A sequence of classic works [23], [24], [25], [26] have considered the case of high-resolution entropy coded quantization with infinite constellation, and subsequent work restricted attention to the case where the infinite constellation is taken as a lattice [27], [28], [20]. Those works considered the MSE case with $\Sigma_X = I_n$, but their conclusion adapted to the WMSE case is that for high-resolution quantization and large n the tradeoff between

distortion and average quantization rate also obeys (16) with $\mathrm{Vol}^{\frac{2}{n}}(\mathcal{S})$ replaced by $2^{2h(W)} = 2\pi e \sigma_W^2$, where $h(\cdot)$ denotes differential entropy. Namely, for entropy coded high-resolution lattice quantization

$$D \approx \left(\sigma^2(UL) \cdot \gamma^{\frac{2}{n}}(UL)\right) 2\pi e \cdot |\Sigma_X|^{\frac{1}{n}} \sigma_W^2 \cdot 2^{-2R}. \tag{17}$$

Since rate and normalized lattice point density are equivalent under either shaped or entropy coded lattice quantization, it suffices to consider the tradeoff between WMSE distortion and $\gamma(L)$.

To that end, we now provide a lower bound on $\sigma^2(UL)$ that depends only on $|U|$ and on $\gamma(L)$. This bound is due to Zador [25] (See also [29, eq. (82)]). It follows from the fact that $\sigma^2(UL)$ is the power of a random vector uniformly distributed over $\mathcal{V}_{UL} \subset \mathbb{R}^n$, which is a convex body of volume $V = |U| \cdot \gamma^{-1}(L)$. Among all bodies in $\mathbb{R}^n$ of volume V, the power of a uniform random vector is minimized when the body is a ℓ_2 ball.[1]

Proposition 1 (Zador): For any full-rank lattice $L \subset \mathbb{R}^n$ and any full-rank matrix $U \in \mathbb{R}^{n\times n}$

$$\sigma^2(UL) \geq \frac{\Gamma\left(\frac{n}{2}+1\right)}{(n+2)\pi} \gamma(L)^{-\frac{2}{n}} |U|^{\frac{2}{n}} \approx \frac{\gamma(L)^{-\frac{2}{n}} |U|^{\frac{2}{n}}}{2\pi e}. \tag{18}$$

In (18), $\Gamma(\cdot)$ is the gamma function. The last approximation can be replaced with a lower bound at the expense of multiplying the right-hand side term by a factor of $\frac{n}{n+2}$. Note that substituting (18) in (17), gives $D \gtrsim D^*_{\mathrm{High-Rate}}(R)$, where $D^*_{\mathrm{High-Rate}}(R)$ is given in (5).

Even without relying on the validity of (17), Proposition 1 provides a simple lower bound on the WMSE distortion of any high-resolution quantization scheme for which the reconstruction satisfies $\hat{W} \in L$, for a lattice $L \subset \mathbb{R}^n$. This follows since for any such quantization scheme we have (due to (7))

$$D \geq \frac{1}{n} \min_{c\in L} \mathbb{E}\|Y - U\cdot c\|^2 = \frac{1}{n}\|Y - Q_{UL}(Y)\|^2,$$

where $Q_{UL} : \mathbb{R}^n \to UL$ is the nearest neighbor (optimal) quantizer for the lattice $UL \subset \mathbb{R}^n$. Under the high-resolution assumption $e_Y = Y - Q_{UL}(Y) \sim \mathrm{Uniform}(\mathcal{V}_{UL})$ and consequently

$$D \geq \sigma^2(UL) \gtrsim \frac{\gamma(L)^{-\frac{2}{n}} |U|^{\frac{2}{n}}}{2\pi e} = |\Sigma_X|^{\frac{1}{n}} \frac{\gamma(L)^{-\frac{2}{n}}}{2\pi e}, \tag{19}$$

where we have used Proposition 1 and the fact that $|U|^2 = |\Sigma_X|$.

For example, in GPTQ/LDLQ it is common to take $L = \alpha\mathbb{Z}^n$ as the base lattice, whose density is $\gamma(L) = \alpha^{-n}$. From (19) we see that its attained distortion must satisfy $D \gtrsim \alpha^2 \frac{|\Sigma_X|^{1/n}}{2\pi e}$.

Note that (19) relies on the assumption $e = UW - Q_{UL}(UW) \sim \mathrm{Uniform}(\mathcal{V}_{UL})$. This holds asymptotically in the limit of high resolution ($\gamma(L) \to \infty$) for all full-rank matrices $U \in \mathbb{R}^{n\times n}$. If one uses dithered lattice quantization [20], this assumption holds exactly for all $\gamma(L)$. However, in the case of dithered lattice quantization the resulting estimate $\hat{W}^*$ would not be in L. Instead, under dithered lattice quantization we have $\hat{W}^* = Y + e$ where $e \sim \mathrm{Uniform}(\mathcal{V}_{UL})$, $e \perp\!\!\!\perp Y$. Consequently, one can benefit by setting $\hat{W} = \beta\hat{W}^*$ with $\beta < 1$ as the estimate for W, as in this case

$$\begin{aligned}\frac{1}{n}\mathbb{E}(X^T(W-\hat{W}))^2 &= \mathbb{E}((1-\beta)X^\top W - \beta e)^2 \\ &= (1-\beta)^2\sigma_W^2\frac{\mathrm{tr}(\Sigma_X)}{n} + \beta^2\sigma^2(UL). \end{aligned}\tag{20}$$

This is the same shrinkage effect that we discussed in [1, Section I.b] Note that $\beta \to 1$ as $\sigma^2(UL)$ decreases, which is the case for high-resolution quantization.

C. Product Codebooks, Codebook Spacing, and Successive Cancellation

This section discusses efficient quantization schemes via successive interference cancellation (SIC). The discussion is not restricted to lattice quantizers, and we will see that SIC can be applied for any product code. Afterwards, we will specialize the discussion to lattice quantizers.

In Section II we argued that the relevant setup for LLMs is the uninformed/Σ_X-oblivious decoder case. We introduced there the problem of quantizing a single column of the weight matrix. In practice, however, a weight matrix will typically have $a \gg 1$ column vectors, and the WMSE matrix Σ_X is common to all of them (as they all operate on the same activations). If the number of column vectors is of the same order as n, it is possible for the decoder to send an $O(n)$ bits description of Σ_X to the uninformed decoder with only a negligible effect on the quantization rate. In particular, the encoder can apply a diagonal matrix $\mathcal{A} = \mathrm{diag}(\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^{n\times n}$, whose spacing coefficients $\{\alpha_i\}$ are chosen based on Σ_X, and quantize W using the codebook $\mathcal{A}\cdot\mathcal{C}$ rather than the codebook $\mathcal{C}$. The cost of reporting $(\alpha_1, \ldots, \alpha_n)$ to the decoder, is amortized over the a columns of the weight matrix, which makes it negligible provided that a is not too small. In fact, spacing the codebook $\mathcal{C}$ used for the reconstruction of W to the codebook $\mathcal{A}\cdot\mathcal{C}$ is

[1] In fact, it is known [30] that for almost all lattices (with respect to the natural measure on the space of lattices) $\sigma^2(L)$ is only $(1 + O(1/n))$ greater than the second moment of the corresponding ℓ_2 ball. Thus, this lower bound is asymptotically attained by a "typical" lattice.

equivalent to changing $X^\top$ to $X^\top \mathcal{A}$. Consequently, in LLMs it is sometimes the case that $\mathcal{A}$ can be absorbed in the activations using the layer-norm with no additional cost associated with describing the scales [11].

Our focus is on obtaining efficient approximate solution to the optimization in equation (6), which becomes equivalent to the optimization in (7) when $\Sigma_X = U^\top U$ is decomposed using the Cholesky decomposition. Assume the codebook $\mathcal{C} \subset \mathbb{R}^n$ is a *product codebook* of the form

$$\mathcal{C} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_n, \quad \text{where } \mathcal{C}_i \subset \mathbb{R} \ \forall i \in [n]. \tag{21}$$

Product codebooks are a common practical choice as it allows for fast (and parallel) decoding. Under the MSE criterion (corresponding to $\Sigma_X = I_n$) they also result in fast optimal encoding, as in this case $U = I_n$ and (7) becomes

$$\begin{aligned} \hat{W}_{\text{MSE}} &= \underset{c_1 \in \mathcal{C}_1, \ldots, c_n \in \mathcal{C}_n}{\operatorname{argmin}} \sum_{i=1}^{n} (W_i - c_i)^2 \\ \implies \hat{W}_{\text{MSE},i} &= \underset{c_i \in \mathcal{C}_i}{\operatorname{argmin}} (W_i - c_i)^2, \quad \forall i \in [n]. \end{aligned} \tag{22}$$

Under the WMSE criterion, corresponding to non-diagonal Σ_X, product codebooks do not in general lend themselves to fast optimal encoding algorithms. In particular, if $\mathcal{C} = \mathbb{Z}^n = \mathbb{Z} \times \cdots \times \mathbb{Z}$, then exactly solving the optimization in (7) requires solving the closest vector problem (CVP) in the lattice $\tilde{L} = U\mathbb{Z}^n$, as observed in [18], [19]. This problem is known to be NP-hard. Consequently, one must resort to sub-optimal algorithms.

Here, we restrict attention to successive interference cancellation (SIC). The most general form of this algorithm is provided in Algorithm 1 and illustrated in Figure 2. In *generalSIC* the n codebooks $\mathcal{C}_1, \ldots, \mathcal{C}_n \subset \mathbb{R}$ whose product is $\mathcal{C} \subset \mathbb{R}^n$ can be arbitrary, and each of them is further scaled by the corresponding α_i. This scaling can in general be absorbed in the codebooks definition. However, since we allow $\mathcal{A}$ to depend on Σ_X through the matrix $U \in \mathbb{R}^{n \times n}$ while the codebooks $\mathcal{C}_1, \ldots, \mathcal{C}_n$ are not allowed to depend on Σ_X, we do not absorb $\mathcal{A}$ into the codebooks. In generalSIC, we first filter the vector Y using the feedforward filter $F \in \mathbb{R}^{n \times n}$. Any matrix $F \in \mathbb{R}^{n \times n}$ can be used here. Then, the vector $\hat{W}$ is generated sequentially, starting from $\hat{W}_n$ up to $\hat{W}_1$. At every step i the scalar $(FY)_i$ is fed to the quantizer to generate $\hat{W}_i$, and then the vector FY is updated to $FY - \hat{W}_i B_{:,i}$. The feedback matrix $B \in \mathbb{R}^{n \times n}$ is strictly lower triangular, due to causality. We have that $\hat{W} = \mathcal{A}Q\left(\mathcal{A}^{-1}(FY - B\hat{W})\right)$. If the quantizer is modeled as adding independent quantization noise Z, we therefore obtain

$$\hat{W} = (I_n + B)^{-1} FY + (I_n + B)^{-1} \mathcal{A} Z. \tag{23}$$

Finally, the estimate $\hat{W}$ is further scaled by $\beta > 0$.

Let us temporarily fix the spacing matrix $\mathcal{A}$. The choices of F and B and β, should strike a balance between two quantities. First, we want the ℓ_2 norm of

$$\begin{aligned} e_Y &= Y - \beta U \hat{W} \\ &= \left(I_n - \beta U (I_n + B)^{-1} F\right) Y - \beta U (I_n + B)^{-1} \mathcal{A} Z \end{aligned} \tag{24}$$

to be as small as possible. On the other hand, we also want $\hat{W}$ to have differential entropy as small as possible, as this will dictate the volume of the shaping region required for capturing it (or equivalently, the average rate of encoding the quantizer's output using entropy coding).

However, in the limit of large quantization rate, where the energy of Z vanishes, the optimal choice of F, B, β tends to the solution for which $\left(I_n - \beta U (I_n + B)^{-1} F\right) = 0$. This corresponds to choosing

$$\begin{aligned} \beta_{\text{High-Rate}} = 1, \ F_{\text{High-Rate}} &= (\operatorname{diag}(U))^{-1}, \\ B_{\text{High-Rate}} &= (\operatorname{diag}(U))^{-1} U - I_n. \end{aligned} \tag{25}$$

If we use the generalSIC algorithm with F, B, β from (25), and use the same scalar codebook $\mathcal{C}_0$ for quantizing all coordinates, the algorithm simplifies to Algorithm 2 which we simply call the SIC algorithm.

As we shall see below, the choice of $\mathcal{A}$ has an important impact on the scheme's performance. In the special case where $\mathcal{A} = \alpha I_n$ for some $\alpha > 0$, the SIC algorithm becomes completely equivalent to the canonical GPTQ algorithm (which in turn, is equivalent to the LDLQ algorithm [17]). This equivalence was recently shown in [18], [19]. Thus, in the sequel, we refer to SIC with $\mathcal{A} = \alpha I_n$ as the GPTQ algorithm. Note however that the GPTQ algorithm was originally presented in [4] from a *noise-shaping* point of view, where the quantization noise is filtered and fed to the next quantizer. In the SIC point of view, it is the signal Y, rather than the quantization noise, that is filtered and fed to the next quantizer.

We find the SIC point of view of [18], [19] more intuitive than the original noise-filtering perspective. Furthermore, under the SIC point of view, the problem is also similar to V-BLAST decoding for the Gaussian MIMO channel [31].

In practice, the sub-codebook $\mathcal{C}_0$ is often taken as the constellations corresponding to data-type FP8, FP4, INT8, etc., resulting in $\hat{W}$ that can be used in fast MatMul hardware. For the remainder of our discussion we will assume $\mathcal{C}_0 = \mathbb{Z}$, such that the product codebook is a lattice. In this case, the SIC algorithm is merely a low-complexity sub-optimal algorithm for the closest vector problem, which is often also referred to as Babai's nearest plane algorithm [32]. For small n the sphere-decoder [33], [34] can be used, and for large n other

Algorithm 1 generalSIC

Inputs: $Y \in \mathbb{R}^n$, feed-forward filter $F \in \mathbb{R}^{n\times n}$, strictly upper triangular feedback filter $B \in \mathbb{R}^{n\times n}$, diagonal spacing matrix $\mathcal{A} = \mathrm{diag}(\alpha_1, \ldots, \alpha_n) \in \mathbb{R}_+^{n\times n}$, scaling coefficient $\beta > 0$ and codebooks $\mathcal{C}_1, \ldots, \mathcal{C}_n \subset \mathbb{R}$
Outputs: $\hat{W} \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_n$.

$Y \leftarrow FY$
for $i = n : 1$ **do**
  $\hat{W}_i \leftarrow \alpha_i Q_i\left(\frac{Y_i}{\alpha_i}\right)$ ▷ $Q_i(x) = \mathrm{argmin}_{c_i \in \mathcal{C}_i}(x - c_i)^2$
  $Y \leftarrow Y - \hat{W}_i \cdot B_{:,i}$ ▷ $B_{:,i}$ is the ith column of B
end for
$\hat{W} \leftarrow \beta \hat{W}$

Algorithm 2 SIC

Inputs: $Y \in \mathbb{R}^n$, upper triangular $U \in \mathbb{R}^{n\times n}$, diagonal matrix $\mathcal{A} = \mathrm{diag}(\alpha_1, \ldots, \alpha_n) \in \mathbb{R}_+^{n\times n}$ and scalar codebook $\mathcal{C}_0 \subset \mathbb{R}$
Outputs: $c_{\mathrm{SIC}} \in \mathcal{C}_0^{\otimes n}$

for $i = n : 1$ **do**
  $c_{\mathrm{SIC},i} \leftarrow \alpha_i Q\left(\frac{Y_i}{\alpha_i U_{i,i}}\right))$ ▷ $Q(x) = \mathrm{argmin}_{c \in \mathcal{C}_0}(x - c)^2$
  $Y \leftarrow Y - c_{\mathrm{SIC},i} \cdot U_{:,i}$ ▷ $U_{:,i}$ is the ith column of U
end for

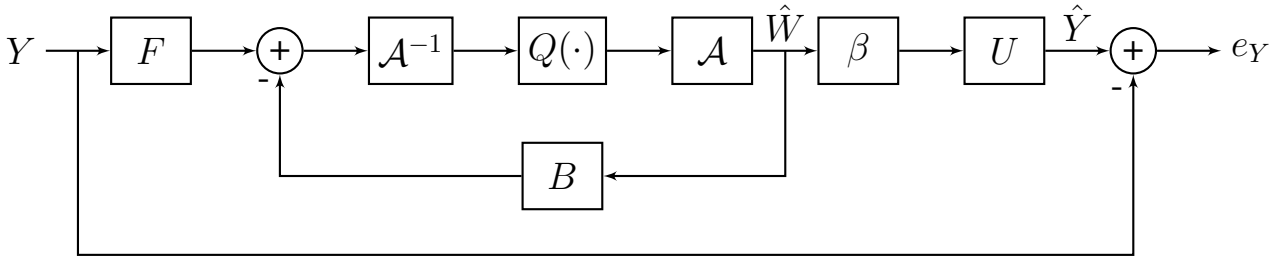


Fig. 2: Illustration of the generalSIC quantization algorithm. The matrix $F \in \mathbb{R}^{n\times n}$ is unrestricted, whereas the matrix $B \in \mathbb{R}^{n\times n}$ must be strictly upper triangular due to causality. The matrix $\mathcal{A} \in \mathbb{R}^{n\times n}$ is positive diagonal, and determines the spacing of the quantizer in each coordinate. The parameter $\beta > 0$ is a scaling parameter. The GPTQ algorithm is obtained as a special case for B, F, β taken as in (25), and $\mathcal{A} = \alpha I_n$. The WaterSIC algorithm is obtained with the same F, B, β, but with $\mathcal{A} = \mathrm{diag}(\alpha_1^{\text{Water}}, \ldots, \alpha_n^{\text{Water}})$ and α_i^{Water} are given in (29)

sub-optimal algorithms can equally be used instead of SIC.

We now analyze the distortion attained by the SIC algorithm, with $\mathcal{C}_0 = \mathbb{Z}$. The resulting $\hat{W}$ is in the lattice $\mathcal{A}\mathbb{Z}^n$, whose cardinality is unbounded. The description of $\hat{W}$ in bits will be handled via shaping or entropy coding, and for now we only constrain the point density of $L = (\alpha_1\mathbb{Z}) \times \cdots \times (\alpha_n\mathbb{Z})$ to $\gamma(L)^{-1} = (\prod_{i=1}^n \alpha_i) = \alpha^n$. Recall that from Proposition 1 and (19) we have that even if the optimal solution to (7) is found, the distortion must satisfy $D \gtrsim \alpha^2 \frac{|\Sigma_X|^{2/n}}{2\pi e}$. Note that for $\mathcal{C}_0 = \mathbb{Z}$, the quantizer Q used in the SIC algorithm takes the simple form $Q(x) = \mathrm{round}(x)$ so that the execution of the SIC algorithm is particularly simple. The next result can be deduced from [32], and we bring the simple proof for completeness.

Lemma 1: Assume we apply the SIC Algorithm with $y \in \mathbb{R}^n$, upper triangular $U \in \mathbb{R}^{n\times n}$, and $\mathcal{C}_i = \alpha_i\mathbb{Z}$ for all $i \in [n]$. Then

$$e_{\mathrm{SIC}} = y - U \cdot c_{\mathrm{SIC}} \in \mathcal{P}_{\{U_{i,i},\alpha_i\}_{i=1}^n},$$

where

$$\mathcal{P}_{\{U_{i,i},\alpha_i\}_{i=1}^n} = \prod_{i=1}^n \left[-\frac{|\alpha_i U_{i,i}|}{2}, \frac{|\alpha_i U_{i,i}|}{2}\right).$$

If in addition $e_{\mathrm{SIC}} \sim \mathrm{Uniform}(\mathcal{P}_{\{U_{i,i},\alpha_i\}_{i=1}^n})$ we have that

$$\begin{aligned} D_{\mathrm{SIC}} &\triangleq \frac{1}{n}\mathbb{E}\|Y - U \cdot c_{\mathrm{SIC}}\|^2 \\ &= \frac{1}{12} \cdot \frac{1}{n}\sum_{i=1}^n (\alpha_i U_{i,i})^2. \end{aligned} \tag{26}$$

Proof. Let $c_{\rm SIC}(y,U)$ be the result of applying the SIC algorithm with inputs $y \in \mathbb{R}^n$ and $U \in \mathbb{R}^{n\times n}$. Denote $\mathcal{A} = \mathrm{diag}(\alpha_1,\ldots,\alpha_n) \in \mathbb{R}^{n\times n}$. The lemma follows from combining the two following observations:

1) $\mathcal{P}_{\{U_{i,i},\alpha_i\}_{i=1}^n} = \{y \in \mathbb{R}^n \ : \ c_{\rm SIC}(y,U) = 0\}$;
2) For any $z \in \mathbb{Z}^n$ it holds that

$$c_{\rm SIC}(y + U\mathcal{A}z, U) = \mathcal{A}z + c_{\rm SIC}(y,U).$$

Thus the SIC algorithm induces the partition of space to decision regions

$$\mathcal{D}_z = U\mathcal{A}z + \mathcal{P}_{\{U_{i,i},\alpha_i\}_{i=1}^n}, \ \forall z \in \mathbb{Z}^n \qquad (27)$$

and $c_{\rm SIC}(y,U) = \mathcal{A}z$ iff $y \in \mathcal{D}_z$. Consequently, $e_{\rm SIC} = y - U\cdot c_{\rm SIC}(y,U) \in \mathcal{P}_{\{U_{i,i},\alpha_i\}_{i=1}^n}$. The claim on $D_{\rm SIC}$ immediately follows from the product structure of $\mathcal{P}_{\{U_{i,i},\alpha_i\}_{i=1}^n}$. ■

It follows from (26) that the distortion for the GPTQ algorithm followed by entropy coding is as given in (10).

By the arithmetic-mean geometric-mean inequality (AM-GM) and (26) we have

$$\begin{aligned} D_{\rm SIC} &= \frac{1}{12}\cdot\frac{1}{n}\sum_{i=1}^n(\alpha_i U_{i,i})^2 \\ &\geq \frac{1}{12}\left(\prod_{i=1}^n \alpha_i\right)^{\frac{2}{n}}\left(\prod_{i=1}^n U_{i,i}\right)^{\frac{2}{n}} = \frac{\alpha^2}{12}|\Sigma_X|^{\frac{2}{n}}. \end{aligned} \qquad (28)$$

The inequality above is achieved with equality iff the scaling factors are of the form

$$\alpha_i^{\rm Water} \triangleq \alpha\cdot\frac{|U|^{\frac{1}{n}}}{|U_{i,i}|}, \quad \forall i \in [n]. \qquad (29)$$

D. WaterSIC algorithm

Motivated by (28) and (29), we propose Algorithm 3: the WaterSIC weight-only quantization algorithm for a weight matrix $W \in \mathbb{R}^{n\times a}$. The properties of the resulting reconstruction $\hat{W}$ are provided in Lemma 2.

Lemma 2: The reconstruction $\hat{W} = \mathrm{diag}(\alpha_1,\ldots,\alpha_n)Z_{\rm SIC}$ produced by the WaterSIC weight-only quantization algorithm satisfies:

1) Any column $\hat{W}_{:,j}$ of $\hat{W}$ belongs to the lattice $L = (\alpha_1\mathbb{Z})\times\cdots\times(\alpha_n\mathbb{Z})$ whose density is $\gamma(L) = \alpha^{-n}$
2) $U(W-\hat{W}) \in \alpha|\Sigma_X|^{1/2n}\cdot\left[-\frac{1}{2},\frac{1}{2}\right)^{n\times a}$
3) If we further assume $U(W-\hat{W}) \sim \mathrm{Uniform}\left(\alpha|\Sigma_X|^{1/2n}\cdot\left[-\frac{1}{2},\frac{1}{2}\right)^{n\times a}\right)$ then

$$D_{\rm WaterSIC} = \frac{1}{an}\mathbb{E}\|X^\top(W-\hat{W})\|^2 = \frac{\alpha^2|\Sigma_X|^{1/n}}{12}.$$

4) $\hat{W} \in W + \alpha|\Sigma_X|^{1/2n}\cdot U^{-1}\left[-\frac{1}{2},\frac{1}{2}\right)^{n\times a}$
5) $Z_{\rm SIC} \in \mathcal{A}^{-1}W + \tilde{U}\left[-\frac{1}{2},\frac{1}{2}\right)^{n\times a}$ where $\mathcal{A} = \mathrm{diag}(\alpha_1,\cdots,\alpha_n)$ and $\tilde{U} = \alpha|\Sigma_X|^{1/2n}\cdot(U\mathcal{A})^{-1}$ is an upper triangular matrix with unit determinant.

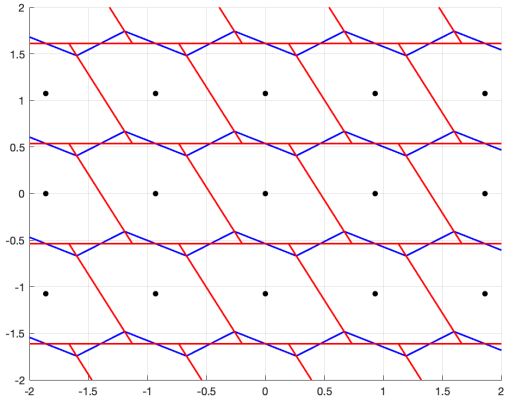


Fig. 3: Illustration of the quantization regions for the optimal lattice quantizer (blue) and for the WaterSIC lattice quantizer (red) for $\Sigma_X = V\Lambda V^T$ where $V = [1\ 1; 1\ -1]/\sqrt{2}$ and $\Lambda = \mathrm{diag}(3,1)$. The lattice used for quantization is $L = (\alpha_1\mathbb{Z})\times(\alpha_2\mathbb{Z})$ where α_1,α_2 are determined via (29), with $\alpha = 1$.

The proof is immediate in light of Lemma 1. Using (15), we see that if the resulting output $Z_{\rm SIC}$ is encoded to bits via ball-shaping or entropy coding, the relation between α and quantization rate R is

$$\alpha \approx \sqrt{2\pi e\sigma_W^2 2^{-2R}}. \qquad (30)$$

Substituting this into item 3 of Lemma 2, gives (11). See [7, Theorem 3.3] for a precise statement. Remarkably, the waterfilling choice of scaling factors (29) result in a lattice whose distortion under the very simple SIC algorithm is only a factor of $\frac{2\pi e}{12} \approx 1.4233$ from the lower bound (19). The loss of this factor is due to the fact that, while the weighted error $U(W-\hat{W})$ lies within a convex set in isotropic position, this set is a cube. The lower bound (19) on the other hand is achieved with equality only if the convex set supporting the weighted error is a ℓ_2-ball. The decision regions for the WaterSIC algorithm, as well as for the optimal lattice quantizer corresponding to $L = (\alpha_1\mathbb{Z})\times(\alpha_2\mathbb{Z})$ are illustrated in Figure 3.

Shaping: There are several ways to represent $Z_{\rm SIC}$ in bits. One way that is simple, but quite inefficient, is using the shaping region

$$\begin{aligned} \mathcal{S}_{\rm rect} &= [-q_1,q_1]\times\cdots\times[-q_n,q_n] \\ &\text{where } q_i = \|Z_{{\rm SIC}i,:}\|_\infty, \quad \forall i\in[n]. \end{aligned} \qquad (31)$$

By definition, all entries of the ith row of $Z_{\rm SIC}$ are in $[-q_i,q_i]$, so overload never occurs. Furthermore, the encoder can describe the shaping region to the decoder by sending $\{q_i\}_{i=1}^n$, with negligible rate if the number of rows is large. And the quantization rate using $\tilde{S}_{\rm rect}$ is

Algorithm 3 WaterSIC weight-only quantization

Inputs: $W \in \mathbb{R}^{n\times a}$, PSD matrix Σ_X and point density $\alpha > 0$.
Outputs: $Z_{\rm SIC} \in \mathbb{Z}^{n\times a}$ and $(\alpha_1, \dots, \alpha_n) \in \mathbb{R}_+^n$ such that $\hat{W} = \mathrm{diag}(\alpha_1, \dots, \alpha_n) Z_{\rm SIC}$.

Compute upper-triangular $U \in \mathbb{R}^{n\times n}$ such that $\Sigma_X = U^\top U$ ▷ Using the Cholesky decomposition
$Y \leftarrow UW$
$\alpha_i \leftarrow \alpha \cdot \frac{|U|^{\frac{1}{n}}}{|U_{i,i}|}, \quad \forall i \in [n]$
$Z_{\rm SIC} \leftarrow 0^{n\times a}$ ▷ Initialize $Z_{\rm SIC}$ with zeros
for $i = n : 1$ **do**
  $Z_{{\rm SIC},i,:} \leftarrow \mathrm{round}\left(\frac{Y_{i,:}}{\alpha_i U_{i,i}}\right)$ ▷ $Y_{i,:}$ and $Z_{{\rm SIC},i,:}$ is the ith row of Y and $Z_{\rm SIC}$, respectively
  $W \leftarrow Y - \alpha_i U_{:,i} \cdot Z_{{\rm SIC},i,:}$ ▷ $U_{:,i}$ is the ith column of U
end for

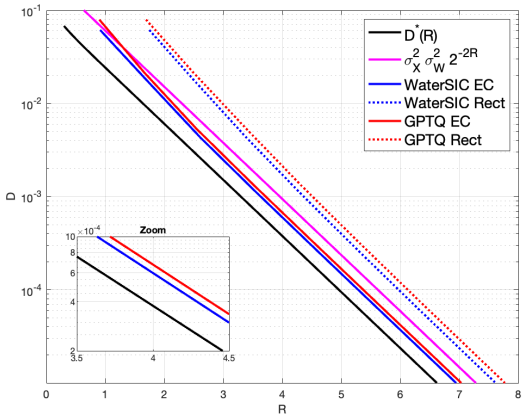


Fig. 4: Performance of several weight-only quantization schemes against benchmark. Here $W \sim \mathcal{N}(0, I_n)$ where $n = 4096$ and Σ_X is the empirical covariance matrix computed from activation samples corresponding to W_v in the 15th layer of Llama3-8B.

$R_{\rm rect} = \frac{1}{n}\sum_{i=1}^n \log(1 + 2q_i)$. Here, we have used the same reconstruction alphabet for an entire row of W. In general, one can decide on different ways to group elements that are described using the same reconstruction alphabet. While using the optimal group size can significantly boost performance, here we do not explore this.

A better, but more complicated, way to perform shaping is via partitioning to cosets, much like in Forney's trellis shaping [35], [27]. In particular, one can use a lattice $L_{\rm Shaping} \subset \mathbb{Z}^n$, with $\mathrm{covol}(L_{\rm Shaping}) = 2^{nR}$, such that $|\mathbb{Z}^n / L_{\rm Shaping}| = 2^{nR}$. In other words, $L_{\rm Shaping}$ partitions $\mathbb{Z}^n$ to 2^{nR} cosets $\{z_i + L_{\rm Shaping}\}$, where $z_1, \dots, z_{2^{nR}} \in \mathbb{Z}^n$ are coset representatives. In order to encode $Z_{{\rm SIC},:,j}$ (the jth column of $Z_{\rm SIC}$) the encoder describes the coset it belongs to, using nR bits. The decoder outputs the coset member $z \in \mathbb{Z}^n$ for which $\sum_{i=1}^n \alpha_i^2 z_i^2$ is minimal. It is easy to verify that the reconstruction constellation corresponding to this scheme is $\mathbb{Z}^n \cap \tilde{\mathcal{S}}_{\text{Lattice-Shaping}}$ where

$$\tilde{\mathcal{S}}_{\text{Lattice-Shaping}} = \Big\{ y \in \mathbb{R}^n \ : \ \sum_{i=1}^n \alpha_i^2 y_i^2 \le \sum_{i=1}^n \alpha_i^2 (y_i - t_i)^2 \quad \forall t \in L_{\rm Shaping} \Big\}. \tag{32}$$

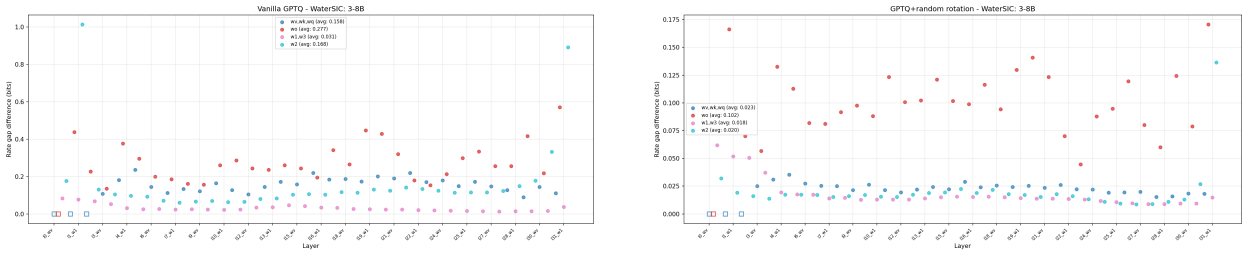
The difficulty here is that the decoder needs to find the coset member that minimizes the scaled energy, which may be computationally intensive unless $L_{\rm Shaping}$ admits special properties. Some progress in finding $L_{\rm Shaping}$ that admit efficient decoding on GPUs was made in [36], [37]. There are many other options one can consider for shaping, which we did not list here.

Finally, the encoder may simply use entropy coding (implemented via any lossless compression package. Note that modern GPU already include dedicated hardware for fast lossless compression/decompression) to describe the entries of $Z_{\rm SIC}$.

E. Quantizing Llama-3-8B

In Figure 4 we plot the rate-distortion curves attained by WaterSIC and GPTQ without tailored spacing (that is, with $\alpha_i = \alpha$) under entropy coding or rectangular shaping corresponding to shaping with $\mathcal{S}_{\rm rect}$ from (31). We see that for high rates the ratio between WaterSIC EC and the fundamental limit $D^*(R)$ from (4) is indeed quite close to $\frac{2\pi e}{12} \approx 1.4233$ (or 0.25 bit in rate). We also see that for this particular Σ_X, at high resolution WaterSIC EC offers only limited gain with respect to GPTQ EC. However, the former is provably close to $D^*(R)$ at high-resolution, whereas the latter is not, and therefore the gap between them may be bigger for other layers, other LLMs, other calibration sets etc.

Let us investigate difference between GPTQ and WaterSIC (entropy coded versions of both) further. Recall that in the high-rate regime, the rate advantage of WaterSIC over GPTQ is estimated as half logarithm of the ratio between arithmetic and geometric mean of squared diagonal entries of U, cf. (10) and (11). We illustrate this



(a) No random rotation.

(b) With random rotation.

Fig. 5: Illustrating rate advantage of WaterSIC over SIC for Σ_X of activations entering various layers of Llama-3-8B when processing Wikitext-2 dataset. Squares correspond to layers with singular Σ_X which we skip. Note that WaterSIC achieves (on random gaussian W) identical performance with or without rotation.

rate advantage on Fig. 5 for all layers of Llama-3-8B. We compare both the case of using (Cholesky decomposition of the) original Σ_X as well as $V^T\Sigma_X V$ for a random orthogonal V. Recall that WaterSIC's performance is independent of rotation, and hence identical in both figures, but the improvement over SIC is much smaller with random rotation. Let us consider implications of this finding.

Cholesky factor interpretation. Recall that k-th diagonal entry $U_{k,k}$ can be interpreted as follows. Think of activations $(X_1,\dots,X_n)$ as a stochastic process. Then, each sample X_k can be decomposed as

$$X_k = X_k^{\|} + X_k^{\perp}\,,$$

where $X_k^{\|}$ is the component that is a linear combination of $(X_1,\dots,X_{k-1})$ (expressible in terms of $U_{k,j}, j = 1,\dots,k-1$) and $X_k^{\perp}$ is the *orthogonal innovation* contained in X_k compared to previous coordinates. $U_{k,k}^2 = \mathbb{E}[(X_k^{\perp})^2]$ is simply the energy of this innovation. Now, applying trace to the identity $\Sigma_X = U^\top U$ we get

$$\sum_{k,j} U_{k,j}^2 = \sum_{j=1}^{n} \lambda_j\,.$$

Therefore, the average $\frac{1}{n}\sum_k U_{k,k}^2$, that governs SIC's performance (10), equals $\frac{1}{n}\sum_j \lambda_j$ minus the sum of squares of off-diagonal entries of U. This implies, that GPTQ's performance is *the worst* in the PCA basis (where U becomes diagonal), which is counter-intuitive. We also see that the gap between D_{iso} (Σ_X-oblivious encoder and decoder) and the waterfilling D^*, which operates in the PCA-basis, is much higher than for GPTQ vs WaterSIC (compare gaps on Fig. 1 vs Fig. 5).

Privileged basis. On the other hand, as we see from comparing two figures on on Fig. 5 applying random rotation clearly improves performance of GPTQ, thus decreasing $\sum_k U_{k,k}^2$. This implies that the original basis is "closer" to the diagonalizing PCA basis than a randomly rotated one. This suggests that the standard basis for X is somehow priviliged among all other possible ones, an effect whose origins are most likely due to operation of Adam [38], which introduces axis aligned outliers in activations.

Cholesky in random basis. Can we estimate $\sum_k U_{k,k}^2$ from knowing the spectrum $\{\lambda_j\}$ of Σ_X? The answer is affirmative if one considers applying random rotation. One such estimate was offered (for a restricted class of Σ_X) by [17, Lemma 2]. However, one can get a more precise information about this sum.

Indeed, when we take V to be a random orthogonal matrix and set $\Sigma_X = V^T\,\mathrm{diag}\{\lambda_j\}V$, quantities $\{U_{k,k}\}$ become functions of a random matrix V. One can show that[2]

$$U_{k,k}^2 = \frac{|\Sigma_X^{(k)}|}{|\Sigma_X^{(k-1)}|}\,,$$

where $|\Sigma_X^{(k)}|$ is a determinant of a $k\times k$ principal minor of Σ_X. In dimensions of interest, these random quantities concentrate quickly around their expectations, which can be computed as[3]

$$\mathbb{E}[|\Sigma_X^{(k)}|] = \frac{1}{\binom{n}{k}}\sum_{|S|=k}\prod_{i\in S}\lambda_i\,.$$

Thus, we can approximate:

$$U_{k,k}^2 \approx \frac{k}{n-k+1}\frac{\sum_{|S|=k}\prod_{i\in S}\lambda_i}{\sum_{|S|=k-1}\prod_{i\in S}\lambda_i}\,. \qquad (33)$$

In particular, $U_{1,1}^2 \approx \frac{1}{n}\sum_{j=1}^n \lambda_j$ and $U_{n,n}^2 \approx \frac{n}{\sum_{j=1}^n \lambda_j^{-1}}$ and the intermediate values smoothly decrease from the arithmetic mean to harmonic mean. This fact is numerically illustrated on Fig. 6. Consequently, performance of

[2]Simply notice that $\Sigma_X^{(k)} = U^{(k)\top}U^{(k)}$.

[3]To see this, let $\tilde V$ be a $n\times k$ matrix consisting of first k columns of V, then from Cauchy-Binet $|\Sigma_X^{(k)}| = |(\Lambda^{\frac12}\tilde V)^\top(\Lambda^{\frac12}\tilde V)| = \sum_{|S|=k}\prod_{i\in S}\lambda_i|\tilde V_{S\times[k]}|^2$. Taking expectation over V we have from symmetry $\mathbb{E}[|\tilde V_{S\times[k]}|^2] = c$ for all subsets $S\subset[n]$. When all $\lambda_i = 1$ we must have $|\Sigma_X^{(k)}| = 1$ and hence $c = \binom{n}{k}^{-1}$.

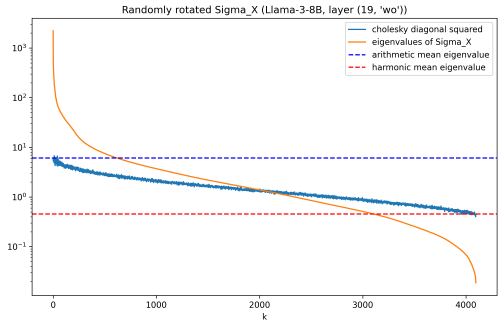


Fig. 6: Illustrating Cholesky diagonals $U_{k,k}^2$ for a randomly rotated $V^\top \Sigma_X V$ and accuracy of approximation (33) in terms of spectrum of Σ_X.

GPTQ with random rotation can be accurately estimated from combining (10) and (33). It is an interesting open problem to estimate worst possible gap (over possible spectra $\lambda_j \geq \epsilon$) between GPTQ with rotation and WaterSIC (which in turn is 0.25-bit away from information-theoretically optimal waterfilling).

WaterSIC vs "universal codebook". We note that theoretical results from [16], discussed in Section II-C, demonstrated that solving (6) optimally essentially attains full waterfilling solution even in low rates, and furthermore does so with a fully Σ_X-oblivious codebook. Note that WaterSIC chooses per-coordinate scales $\alpha_i \propto U_{i,i}^{-1}$ (Cholesky decomposition of) Σ_X and, thus, requires large $a \gg 1$ to amortize sending of α_i's in high precision (usually BF16). The results in [16] on the other hand, are valid for any $a \geq 1$.

F. Future of weight-only quantization

The high-resolution approximations (16) and (17) in Subsection III-B, combined with the analysis of WaterSIC in Subsection III-D, show that when WaterSIC followed by EC or close-to-optimal spherical shaping is used, we attain the optimal information theoretic distortion from (5) up to a multiplicative gap of $\frac{2\pi e}{12}$. Figure 4 numerically confirms the accuracy of the high-resolution approximations. Furthermore, Figure 5 shows that with random rotation even the vanilla GPTQ algorithm, which is completely equivalent to the widely used GPTQ/LDLQ algorithms, achieves distortion only slightly greater than WaterSIC, and is therefore nearly optimal. In light of this, one may wonder whether the practical schemes are already so good that there is no further room for improvement. We claim that this is not the case, and list several important directions for improved quantization schemes:

First, in this survey paper we restricted attention to the high-rate regime. At small quantization rates (say $0.5-2$ bits per entry) many other effects will become important: necessity of Johnson-Lindenstrauss dimensionality reduction, as shown in [2], shrinkage and other low-rate effects. Proper shaping and exploiting structure of W matrices will be paramount. Indeed, at low rates [7] shows WaterSIC to have massive advantage over GPTQ, even if the latter is Huffman coded.

Second, even our analysis for the high-resolution regime relied on applying EC or near-optimal spherical shaping after quantization. While EC over $\mathbb{Z}$ can be efficiently implemented, even on modern GPUs EC decompression is about an order of magnitude slower than loading the uncompressed entries. Thus, the usefulness of EC for running quantized models on a GPU is somewhat questionable. On the other hand, simple shaping methods that can be easily made efficient, e.g. rectangular shaping, may be highly suboptimal (see Figure 4). The problem of shaping schemes that are amenable to fast GPU implementation is at its infancy [12].

Third, while the $\frac{2\pi e}{12}$ high-resolution gap-to-optimality of WaterSIC+EC is relatively small, reducing it further is of interest. This gap stems from the sub-optimality of the integer lattice $\mathbb{Z}^n$ as a quantizer. Generally, we can decrease this gap by combining WaterSIC with jointly quantizing d entries from the same row of W to a lattice $L' \subset \mathbb{R}^d$, followed by EC. If we use the optimal lattice quantizer in $\mathbb{R}^{d'}$ we can reduce the $\frac{2\pi e}{12}$ gap to optimality to $2\pi e \cdot G_d$ where G_d is the optimal normalized second moment in dimension d. See [39, Table I] for a list of the best known NSM for $1 \leq d \leq 48$ as of 2023 (some improvements were reported in certain dimensions since then). The downside of replacing $\mathbb{Z}$ with a higher-dimensional lattice is that EC of the quantizer's output becomes more challenging as its cardinality is much larger.

Fourth, implementation of WaterSIC relies on the fact that sending per-channel scales α_i can be amortized freely, which is only the case when one has many output

neurons in a linear layer (large $a \gg 1$).

ACKNOWLEDGMENT

We thank Alina Harbouzova, Egor Lifar, and Semyon Savkin (MIT EECS) for numerous discussions and help with obtaining calibration matrices for Llama-3-8B.

REFERENCES

[1] O. Ordentlich and Y. Polyanskiy, "High-rate quantized matrix multiplication I," 2026.

[2] ——, "Optimal quantization for matrix multiplication," *arXiv preprint arXiv:2410.13780*, 2024.

[3] M. Nagel, R. A. Amjad, M. Van Baalen, C. Louizos, and T. Blankevoort, "Up or down? adaptive rounding for post-training quantization," in *International conference on machine learning*. PMLR, 2020, pp. 7197–7206.

[4] 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

[5] 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.

[6] B. Hassibi, D. G. Stork, and G. J. Wolff, "Optimal brain surgeon and general network pruning," in *IEEE international conference on neural networks*. IEEE, 1993, pp. 293–299.

[7] E. Lifar, S. Savkin, O. Ordentlich, and Y. Polyanskiy, "Watersic: information-theoretically (near) optimal linear layer quantization," *arXiv preprint arXiv:2603.04956*, 2026.

[8] Y. Li, R. Gong, X. Tan, Y. Yang, P. Hu, Q. Zhang, F. Yu, W. Wang, and S. Gu, "Brecq: Pushing the limit of post-training quantization by block reconstruction," *arXiv preprint arXiv:2102.05426*, 2021.

[9] A. Tseng, Z. Sun, and C. De Sa, "Model-preserving adaptive rounding," *arXiv preprint arXiv:2505.22988*, 2025.

[10] H. Badri and A. Shaji, "Half-quadratic quantization of large machine learning models," November 2023. [Online]. Available: https://mobiusml.github.io/hqq_blog/

[11] 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.

[12] S. Savkin, E. Porat, O. Ordentlich, and Y. Polyanskiy, "NestQuant: Nested lattice quantization for matrix products and LLMs," *arXiv preprint arXiv:2502.09720*, 2025.

[13] S. Zhang, H. Zhang, I. Colbert, and R. Saab, "Qronos: Correcting the past by shaping the future... in post-training quantization," *arXiv preprint arXiv:2505.11695*, 2025.

[14] H. Zhang, S. Zhang, I. Colbert, and R. Saab, "Provable post-training quantization: Theoretical analysis of optq and qronos," *arXiv preprint arXiv:2508.04853*, 2025.

[15] Y. Polyanskiy and Y. Wu, *Information theory: From coding to learning*. Cambridge university press, 2024.

[16] A. Harbuzova, O. Ordentlich, and Y. Polyanskiy, "Price of metric universality in vector quantization is at most 0.11 bit," *arXiv preprint arXiv:2602.05790*, 2026.

[17] J. Chee, Y. Cai, V. Kuleshov, and C. M. De Sa, "Quip: 2-bit quantization of large language models with guarantees," *Advances in Neural Information Processing Systems*, vol. 36, pp. 4396–4429, 2023.

[18] J. Chen, Y. Shabanzadeh, E. Crnčević, T. Hoefler, and D. Alistarh, "The geometry of llm quantization: Gptq as babai's nearest plane algorithm," *arXiv preprint arXiv:2507.18553*, 2025.

[19] J. Birnick, "The lattice geometry of neural network quantization–a short equivalence proof of gptq and babai's algorithm," *arXiv preprint arXiv:2508.01077*, 2025.

[20] R. Zamir, *Lattice Coding for Signals and Networks: A Structured Coding Approach to Quantization, Modulation, and Multiuser Information Theory*. Cambridge University Press, 2014.

[21] 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.

[22] O. Ordentlich, O. Regev, and B. Weiss, "Bounds on the density of smooth lattice coverings," *arXiv preprint arXiv:2311.04644*, 2023.

[23] W. R. Bennett, "Spectra of quantized signals," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 446–472, 1948.

[24] P. Panter and W. Dite, "Quantization distortion in pulse-count modulation with nonuniform spacing of levels," *Proceedings of the IRE*, vol. 39, no. 1, pp. 44–48, 1951.

[25] P. Zador, "Asymptotic quantization error of continuous signals and the quantization dimension," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 139–149, 1982.

[26] A. Gersho, "Asymptotically optimal block quantization," *IEEE Transactions on information theory*, vol. 25, no. 4, pp. 373–380, 1979.

[27] M. V. Eyuboglu and G. D. Forney, "Lattice and trellis quantization with lattice-and trellis-bounded codebooks-high-rate theory for memoryless sources," *IEEE Transactions on Information theory*, vol. 39, no. 1, pp. 46–59, 1993.

[28] R. Zamir and M. Feder, "On lattice quantization noise," *IEEE Transactions on Information Theory*, vol. 42, no. 4, pp. 1152–1159, 1996.

[29] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, 3rd ed., ser. Grundlehren der mathematischen Wissenschaften. New York: Springer-Verlag, 1999, vol. 290.

[30] O. Ordentlich, "The Voronoi spherical cdf for lattices and linear codes: New bounds for quantization and coding," *arXiv preprint arXiv:2506.19791*, 2025.

[31] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-blast: An architecture for realizing very high data rates over the rich-scattering wireless channel," in *1998 URSI international symposium on signals, systems, and electronics. Conference proceedings (Cat. No. 98EX167)*. IEEE, 1998, pp. 295–300.

[32] L. Babai, "On lovász'lattice reduction and the nearest lattice point problem," *Combinatorica*, vol. 6, no. 1, pp. 1–13, 1986.

[33] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of computation*, vol. 44, no. 170, pp. 463–471, 1985.

[34] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE transactions on information theory*, vol. 48, no. 8, pp. 2201–2214, 2002.

[35] G. D. Forney, "Trellis shaping," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 281–300, 1992.

[36] S. Savkin, E. Porat, O. Ordentlich, and Y. Polyanskiy, "Nestquant: Nested lattice quantization for matrix products and llms," *Proc. International Conference on Machine Learning (ICML)*, 2025.

[37] A. Tseng, Q. Sun, D. Hou, and C. De Sa, "Qtip: Quantization with trellises and incoherence processing," *arXiv preprint arXiv:2406.11235*, 2024.

[38] N. Elhage, R. Lasenby, and C. Olah, "Privileged bases in the transformer residual stream," *Transformer Circuits Thread*, 2023. [Online]. Available: https://transformer-circuits.pub/2023/privileged-basis/index.html

[39] E. Agrell and B. Allen, "On the best lattice quantizers," *IEEE Transactions on Information Theory*, vol. 69, no. 12, pp. 7650–7658, 2023.