Rounding Error Statistics as Numerics Signature

Ping Tak Peter Tang
Riovs Inc., Santa Clara, CA 95954

***Abstract*—The current proliferation of application specific computer processors brings us many customized computer arithmetic designs with various precisions and/or numerical properties. Increasingly, application developers find themselves carrying out numerical quality assurance. The tools for this task are quite limited. A common method that requires computed result of a kernel operation on an example input to be accurate within a certain threshold is not robust: Not only is a good threshold hard to establish, but this very approach is fundamentally unreliable, as computation errors are so sensitive to minute computational differences that they can be quite different even using numerics of similar qualities. In this paper, we show that the statistics, rather than a few isolated instances, of computation errors are so robust that they can be thought of as signatures of the underlying numerics. We can therefore compared these statistics against reference signatures as a more robust quality assurance, or use observed statistics as diagnostic methods, or incorporate them during numerics and arithmetic design.**

I. Introduction

Due in no small part to the tremendous computation needs of AI, the computer industry is witnessing a proliferation of application specific architectures for accelerators as well as complete systems such as [1], [2], [3] and [4], to cite just a few that are of public knowledge. Their underlying computer arithmetic formats and/or behavior may diverge from the more familiar and standardized IEEE standard. See [5] and [6] for example. Even when some parts of these arithmetic are documented, subtle behavioral idiosyncrasies are often not (see for example [7]). Consequently, software developers increasingly need to install safeguards such as operator tests to ensure the numerical integrity of basic computational blocks such as products of small matrices. These tests are typically single-example threshold based (e.g. made convenient by `torch.testing.assert_allclose`): One example is fed to a computational kernel under test. A measure of the error, that is, the difference between the computed result and a reference is obtained. The test passes if and only if the measure does not exceed a predetermined threshold.

The main problem of this approach is that numerical errors caused by finite-precision arithmetic are very sensitive to low-level details of a numerical processes. Two processes of comparable precision executing the same input can result in noticeably different error measures. So the threshold must not be too stringent. Nevertheless a threshold that allows for the maximum possible error among all acceptable numerical behavior may allow bad numerics to pass as well. Indeed, anecdotes among many colleagues have a common thread: these thresholds have been repeatedly increased to accommodate new test cases that must made pass lest developments of other parts of software projects be blocked.

We make the case here that the errors of a well-defined numerical process on inputs that are also drawn from well-defined probabilistic distributions are themselves well-defined stochastic processes. Therefore, statistics – as opposed to a single sample – of errors can serve as much more reliable thresholds. To support this argument, this paper explicitly computes the variance of rounding errors of several common computation kernels on specific distributions of input values. The sampled variances of experiments match their corresponding theoretical values so well that the former can be considered as signatures of the numerics behind.

In the following, Section II relates our work to recent stochastic approaches in the study of rounding errors. Section III presents the mathematical probabilistic settings and the main tools we developed for subsequent analyses. We then present several examples of computational kernels and scenarios of numerics where we demonstrate that the sampled variances of errors are just as our theoretical analyses predict. Section IV considers summation, Section V considers inner and matrix products, and Section VI considers a special accumulator architecture that uses a mixture of fixed-point and floating-point arithmetic. Section VII summarizes our work and discusses next steps.

II. Related Works

Analysis of the effects on computational tasks due to rounding errors inherent in finite-precision arithmetic was pioneered by Wilkinson (see the recently republished classic text [8]). Higham's book [9] is a more recent standard reference on this subject. These standard works try to establish various worst-case error bounds on a number of algebraic processes such as summations (c.f. [10]). Nevertheless, worst-case errors are in fact very rarely encountered. A body of more recent investigations such as [11], [12], [13], [14] establishes tighter error bounds that are satisfied with high probabilities. Similar to these works and especially to that of [13], we model accumulated rounding errors in algebraic processes as probability distributions. In contrast, however, we focus not on error bounds but on summary statistics of these distributions that can be used as signatures of the numerics behind various computational kernels.

III. Basic Setup

Consider drawing a real value x from a normal distribution with mean 0 and variance σ^2: $x \sim \mathcal{N}(0, \sigma^2)$. We examine the rounding error incurred when x is rounded to IEEE single precision in the round-to-nearest-even mode. The rounding error is $\mathcal{R}_\sigma \stackrel{\text{def}}{=} \mathrm{fl}(x) - x$ where $\mathrm{fl}(x)$ denotes rounding a

2576-2265/25/$31.00 ©2025 IEEE
DOI 10.1109/ARITH64983.2025.00024

value x to a floating-point value. As a random variable, it is reasonable to model $\mathcal{R}_\sigma$ conditioned on $|x|$ falling into the binary interval $I_k \stackrel{\text{def}}{=} [2^k, 2^{k+1})$ by the uniform distribution between $[-2^k\epsilon/2, 2^k\epsilon/2]$ ($\pm$ half a unit-of-last-place, e.g. $\epsilon = 2^{-23}$ for single precision). This is equivalent to assuming the significant bits x from the 25-th bit onward are equally likely to be either 0 or 1. In mathematical notation:

$$\mathcal{R}_\sigma\big|_{|x|\in I_k} \sim \mathcal{U}[-2^k\epsilon/2, 2^k\epsilon/2].$$

The probability density function of $\mathcal{R}_\sigma$ is thus of a staircase shape, the summations of the density functions of $\mathcal{U}[-2^k\epsilon/2, 2^k\epsilon/2]$ weighted by the $p_k(\sigma)$, the probability of $|x| \in I_k$ when $x \sim \mathcal{N}(0, \sigma^2)$ is the definite integral of the normal density function $\rho(t) = e^{-\frac{t^2}{2\sigma^2}}/\sqrt{2\pi\sigma^2}$ on $\pm I_k$, exploiting $\rho(-t) = \rho(t)$:

$$p_k(\sigma) \stackrel{\text{def}}{=} \frac{2}{\sqrt{2\pi\sigma^2}} \int_{2^k}^{2^{k+1}} e^{\frac{-t^2}{2\sigma^2}}\, dt. \tag{1}$$

This density function of $\mathcal{R}_\sigma$ and a histogram of rounding 10^5 double-precision $x \sim \mathcal{N}(0, 1)$ to single precision is shown in Figure 1.

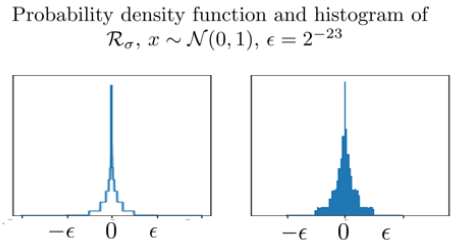

Fig. 1. Error when rounding to FP32: left is the model density function; right is experimental data

Let us obtain the second and fourth moments of $\mathcal{R}_\sigma$. The second and fourth moments of the uniform distribution $\mathcal{U}[-1/2, 1/2]$ are easily obtained as $1/12$ and $1/80$, respectively. For all practical purposes, $p_k(\sigma)$ is negligible beyond the exponent range of single precision numbers. We define $F(\sigma)$ and $G(\sigma)$ for ease of subsequent presentations via these equalities (using $E(\cdot)$ as expectation):

$$E(\mathcal{R}_\sigma^2) = \frac{\epsilon^2}{12} \sum_{k=-\infty}^{\infty} 2^{2k} p_k(\sigma) \stackrel{\text{def}}{=} \frac{\epsilon^2}{12} F(\sigma). \tag{2}$$

Similarly,

$$E(\mathcal{R}_\sigma^4) = \frac{\epsilon^4}{80} \sum_{k=-\infty}^{\infty} 2^{4k} p_k(\sigma) \stackrel{\text{def}}{=} \frac{\epsilon^4}{80} G(\sigma). \tag{3}$$

It turns out one can obtain rather economically the numerical values of $E(\mathcal{R}_\sigma^2)$ and $E(\mathcal{R}_\sigma^4)$ at a general σ. First, note that $p_k(2\sigma) = p_{k-1}(\sigma)$ by a simple change-of-variable integration on Equation 1. Consequently,

$$F_0(\sigma) \stackrel{\text{def}}{=} F(\sigma)/\sigma^2 \quad \text{and} \quad G_0(\sigma) \stackrel{\text{def}}{=} G(\sigma)/\sigma^4 \tag{4}$$

are defined completely on the interval $[1, 2]$ as $F_0(2\sigma) = F_0(\sigma)$ and $G_0(2\sigma) = G_0(\sigma)$ for all σ. This periodicity property for F_0 and G_0 is easily derived from their definitions and the property $p_k(2\sigma) = p_{k-1}(\sigma)$. Second, both $F_0(\sigma)$ and $G_0(\sigma)$ (cf. Figure 2) can be conveniently computed one time on a dense subset of $[1, 2]$ using readily available software packages such as `scipy` in Python.

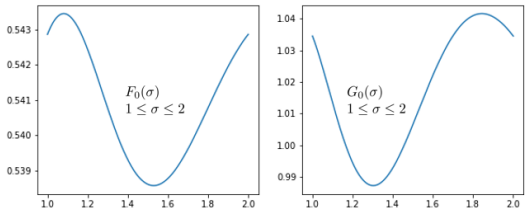

Fig. 2. The periodicized second and fourth moment of $\mathcal{R}_\sigma$, $F_0(\sigma)$ and $G_0(\sigma)$

We can then substitute $F_0(\sigma)$ and $G_0(\sigma)$ by low-degree polynomials via for example a least-squares approximation or a minimax approximation. In our experiments, we used a degree-6 and a degree-8 polynomial for the second and fourth moment, respectively. Finally, given any $\sigma > 0$, $E(\mathcal{R}_\sigma^2) = \frac{\epsilon^2}{12}\sigma^2 F_0(\sigma_0)$ where σ_0 is σ scaled by the integer power of 2 such that $\sigma_0 \in [1, 2)$. Similarly, $E(\mathcal{R}_\sigma^4) = \frac{\epsilon^4}{80}\sigma^4 G_0(\sigma_0)$.

We now enhance our tool set by considering the rounding error in adding two numbers drawn from zero-mean normal distributions of different variances σ_x^2 and σ_y^2. If both addends were real numbers, then the rounding error in summation is $\mathcal{R}_\sigma$ where $\sigma^2 = \sigma_x^2 + \sigma_y^2$ as the sum of two zero-mean normal distributions is the zero-mean normal distribution with variance equals to the sum of the addends' variances. In practice, we will often add floating-point values $x = \text{fl}(\alpha)$ and $y = \text{fl}(\beta)$ where $\alpha \sim \mathcal{N}(0, \sigma_x^2)$, $\beta \sim \mathcal{N}(0, \sigma_y^2)$. For simplicity, we will use the notation $x \sim \text{fl}(\mathcal{N}(0, \sigma_x^2))$ and $y \sim \text{fl}(\mathcal{N}(0, \sigma_y^2))$. The moments of the rounding error random variable in general will be slightly higher: just take for example if we are rounding a number in $[1, 2)$ with only 1 extra bit. The rounding error is $-\epsilon/2, 0, \epsilon/2$ with probabilities $1/4, 1/2, 1/4$, respectively. The second moment is thus $\epsilon^2/8$ and not $\epsilon^2/12$. In some special instances, however, the rounding error will be smaller. For example $-2 \le x/y \le -1/2$ leads to the sum being exact and hence of zero error. Thus, given that the unrounded sum's magnitude of two floating-point numbers is in the binary interval I_k, we refine our model of rounding error by further conditioning on the ratio x/y. Let $\mathcal{S}_{\sigma,r}$ denote the rounding error random variable of adding floating-point numbers in $\mathcal{N}(0, \sigma_x^2)$ and $\mathcal{N}(0, \sigma_y^2)$ where $\sigma^2 = \sigma_x^2 + \sigma_y^2$ and $r = \sigma_x^2/\sigma_y^2$. This gives us a form

$$E(\mathcal{S}_{\sigma,r}^2 \big| |s| \in I_k) = \frac{\epsilon^2}{12} 2^{2k} \left(\sum_{\ell,\pm} \alpha_{\ell,\pm} P(\frac{x}{y} \in \pm I_\ell) \right),$$

The quotient of two normal distribution is a Cauchy distribution [15] distribution and parameterized solely by the ratio

$r = \sigma_x^2/\sigma_y^2$. Consequently, we write

$$E(\mathcal{S}_{\sigma,r}^2) = \frac{\epsilon^2}{12} F(\sigma)\phi(r) \tag{5}$$

where $\sigma^2 = \sigma_x^2 + \sigma_y^2$. It also suffices to consider $\sigma_x^2 \le \sigma_y^2$ as we assume floating-point sum to be commutative. We used a large sample of mean square errors for $E(\mathcal{S}_{\sigma,r}^2)$ to obtain numerical values for $E(\mathcal{S}_{\sigma,r}^2)/F(\sigma)$ at a fixed r, thus arriving at $\phi(r)$. We also verified that this value remains unchanged (up to experimental noise) by changing σ but leaving r fixed. Similarly,

$$E(\mathcal{S}_{\sigma,r}^4) = \frac{\epsilon^4}{80} G(\sigma)\psi(r) \tag{6}$$

Moreover, by computing this ratio for a fixed σ but on a dense set of r values in $[0, 1]$, we obtained low-degree polynomial approximations to $\phi(r)$ and $\psi(r)$ as well.
For completeness, here are the coefficients we used for $F_0(\sigma)$ and $G_0(\sigma)$ computed as $\sum_{j=0}^{\deg} c_j(\sigma-1)^j$ for $\sigma \in [1, 2]$; $\phi(r)$ and $\psi(r)$ are computed as $\sum_{j=0}^{\deg} c_j r^j$.

	$F(\sigma)/\sigma^2$	$G(\sigma)/\sigma^4$	$\phi(r)$	$\psi(r)$
c_0	0.54279	1.03445	1.00684	0.99967
c_1	0.01827	-0.16159	0.88421	1.94703
c_2	-0.13577	-0.85648	-1.91853	-2.43573
c_3	0.16304	4.03253	-1.93557	5.04192
c_4	0.10800	-2.64662	-0.72735	-16.59014
c_5	-0.26253	-7.00256	0.00000	29.25313
c_6	0.10911	13.25400	0.00000	-24.83812
c_7	0.00000	-8.72614	0.00000	9.10214
c_8	0.00000	2.10696	0.00000	-0.84897

Our ability to calculate the second and fourth moments of both $\mathcal{R}_\sigma$ and $\mathcal{S}_{\sigma,r}$ allow us to derive the variance and variance-of-variance of accumulated rounding errors in many controlled experiments. We will also see that the variance of rounding errors can effectively serve as signatures of the underlying numerics that lead to these errors.

IV. Vector Sum

Consider the summation of L elements in a simple recursive manner except with a possible SIMD extension, entirely in IEEE single precision. Specifically, the summation is a recursive accumulation not necessarily directly of the input elements but rather of partial sums of a number of elements equal to the SIMD length. The following pseudo code describes the numerical characteristic of this summation.

```
def sgl_simd_sum(x_data, simd_len):
    L = len(x_data); s = 0.0; ind = 0;
    n_outer = L // simd_len
    for _ in range(n_outer):
        tmp = 0.0
        for _ in range(simd_len):
            tmp = tmp + x_data[ind]
            ind += 1
        s = s + tmp
    return s
```

It is reasonable to expect that summations with different SIMD lengths give rise to different numerical behavior. Our thesis is that statistics such as mean squared errors of computation kernels like this summation can technically serve as signatures of the underlying numerics as long as the inputs are drawn in a controlled manner. To support this thesis, we show here that the total accumulated rounding error in the sum with different SIMD lengths yield distinct differentiating signatures. We analyze the rounding error statistics when the inputs are drawn independently from the normal distribution $\mathcal{N}(0, 1)$. The sampled statistics of different summation configurations are then shown to indeed offer distinct signatures, all of which match the theoretical analysis well.

A. Error Statistics Analysis

Let $\Delta_{(\ell)}$ be the random variable of the summation error with SIMD length ℓ. Assuming ℓ divides L and $m \stackrel{\text{def}}{=} L/\ell$. The summation process consists of m SIMD sum of ℓ elements, followed by $m-1$ summation of the floating-point partial sums. Each length-ℓ SIMD vector sum consists of $\ell - 1$ summations of two floating-point numbers. Thus the entire summation process consists of $m(\ell-1) + m - 1 = L - 1$ summations of a floating-point number pair in $\text{fl}(\mathcal{N}(0, \sigma_x^2))$ and $\text{fl}(\mathcal{N}(0, \sigma_y^2))$ of various values of σ_x^2 and σ_y^2.
Consider first the length-ℓ SIMD sum. Generically, it is of the form

$$a_0 = y_0; \quad a_i = \text{fl}(a_{i-1} + y_i),\ i = 1, 2, \ldots, \ell - 1.$$

Because each $y_i \sim \text{fl}(\mathcal{N}(0, 1))$, the $\ell - 1$ error random variables are modeled by $\mathcal{S}_{\sigma_i,r_i}$ where $\sigma_i = \sqrt{1+i}$, and $r_i = 1/i$, for $i = 1, 2, \ldots, \ell-1$. After the length-ℓ SIMD sums, we have m floating-point numbers z_j, $j = 0, 1, \ldots, m-1$ and the simple recursive sum of these partial sums follows

$$b_0 = z_0; \quad b_i = \text{fl}(b_{i-1} + z_i),\ i = 1, 2, \ldots, m - 1.$$

Because each $z_i \sim \text{fl}(\mathcal{N}(0, \ell))$, the $m - 1$ error random variables are modeled by $\mathcal{S}_{\sigma_i,r_i}$ where $\sigma_i = \sqrt{(i+1)\ell}$ and $r_i = 1/i$, $i = 1, 2, \ldots, m-1$. The total summation error $\Delta_{(\ell)}$ is of the form

$$\Delta_{(\ell)} = \tau_1 + \tau_2 + \cdots + \tau_{L-1}$$

where each τ_j is of the form $\mathcal{S}_{\sigma_j,r_j}$ for the specific σ_j and r_j described above. Furthermore, it is reasonable to assume mutual independence of these τ_js. As IEEE rounding is unbiased, $E(\tau_j) = 0$ and hence

$$\text{Var}(\Delta_{(\ell)}) = \sum_j E(\tau_i^2), \tag{7}$$

which is $\epsilon^2/12$ times

$$m \sum_{i=1}^{\ell-1} F(\sqrt{1+i})\phi(1/i) + \sum_{i=1}^{m-1} F(\sqrt{(1+i)\ell})\phi(1/i). \tag{8}$$

We can also obtain the variance of $\Delta_{(\ell)}^2$. Recall that the analysis on $\Delta_{(\ell)}$ shows that it is the sum of random variables τ_i's each of which is the rounding error in summing two floating-point numbers distributed in $\mathcal{N}(0, \sigma_x^2)$ and $\mathcal{N}(0, \sigma_y^2)$, $\sigma_x^2 \le \sigma_y^2$. We also obtained that $E(\tau_i^2)$ and $E(\tau_i^4)$ as $F(\sigma)\phi(r)$ and

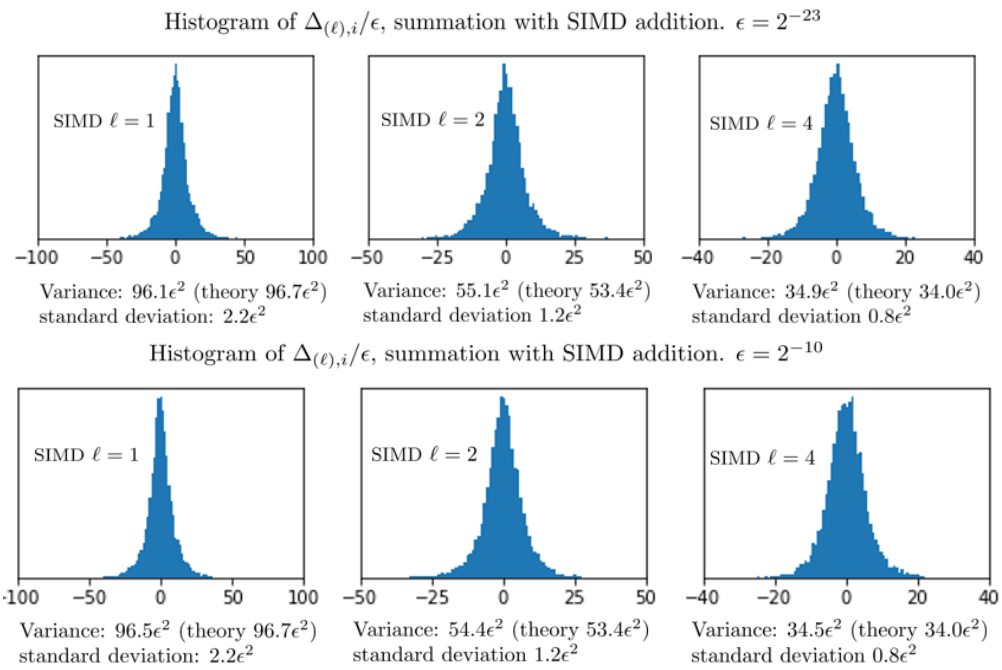

Fig. 3. Illustration of using observed mean square error as a signature of the underlying numerics. Top experiment is in IEEE single precision. The bottom experiment using IEEE FP16.

$G(\sigma)\psi(r)$, respectively, where $\sigma^2 = \sigma_x^2 + \sigma_y^2$ and $r = \sigma_x^2/\sigma_y^2$. By definition,

$$\mathrm{Var}(\Delta_{(\ell)}^2) = E(\Delta_{(\ell)}^4) - E^2(\Delta_{(\ell)}^2).$$

By multinomial expansion, we have

$$E(\Delta_{(\ell)}^4) = \sum_i E(\tau_i^4) + 12\sum_{i<j} E(\tau_i^2)E(\tau_j^2)$$

because expectation terms with odd powers of τ_i are 0. Similarly

$$\begin{aligned} E^2(\Delta_{(\ell)}^2) &= (\sum_i E(\tau_i^2))^2 \\ &= \sum_i E^2(\tau_i^2) + 2\sum_{i<j} E(\tau_i^2)E(\tau_j^2) \end{aligned}$$

Thus

$$\mathrm{Var}(\Delta_{(\ell)}^2) = \sum_i (E(\tau_i^4) - E^2(\tau_i^2)) + 10\sum_{i<j} E(\tau_i^2)E(\tau_j^2). \tag{9}$$

Suppose we have N samples of $\Delta_{(\ell)}^2$: $\Delta_{(\ell),i}^2$, $i = 1, 2, \ldots, N$ and assuming N is large enough that the Central Limit Theorem is applicable, then

$$\frac{1}{N}\sum_{i=1}^{N} \Delta_{(\ell),i}^2 \sim \mathrm{Var}(\Delta_{(\ell)}) + \frac{\sigma}{\sqrt{N}}\mathcal{N}(0,1), \tag{10}$$

where $\sigma^2 = \mathrm{Var}(\Delta_{(\ell)}^2)$. The standard deviation of the sampled variance using N samples is $\sigma/\sqrt{N}$. Thus, we can make an assessment such as: if the underlying sum is using SIMD length of ℓ, then with about 95% chance we expect the observed mean-squared error to be within 2σ of the theoretical variance of the error.

B. Experimental Corroboration

We demonstrate that the experimental data match the theory well and that the former can indeed serve as a signature of its underlying numerics. We set $L = 64$ and for each SIMD length of $\ell = 1, 2, 4$ we draw $N = 10^4$ random length-L vector $\mathbf{x} = [x_0, x_1, \ldots, x_{L-1}]$ each element of which comes from $\mathrm{fl}(\mathcal{N}(0,1))$. The error of the computed sum is simply its difference from an accurate sum obtained by up-conversion of $\mathbf{x}$ to double precision followed by straightforward summation. So for each SIMD length ℓ, we collected N samples $\Delta_{(\ell),i}$, $i = 1, 2, \ldots, N$ of the total accumulated error of the summation process. Figure 3 shows the histograms and relevant metrics of the accumulated error using different SIMD lengths. In particular, the experimental values $\sum_{i=1}^{N} \Delta_{(\ell),i}^2/N$ match the theoretical values given by 8 and they are also well within two standard deviations from each other, which we would expect with high likelihood as suggested by Equation 10.

C. General Summation

As a final example for summation, consider the general summation stated in [9] as Algo 4.1 (Figure 4).

We fix an order of summation by creating an ordered list of $n-1$ pairs of distinct indices (i, j) each from the set

```
S := [x_1, x_2, ..., x_n]
for k = n, n-1, ..., 2
    pick distinct i, j from {1,2.,,,.k}
    a := S[i], b := S[j], c := fl(a+b)
    delete S[i], S[j] from S; append c to S
    // S is now a list of length k-1
return S[1]
```

Fig. 4. General Summation

$\{1, 2, \ldots, k\}$ from $k = n, n-1, \ldots, 2$. The numerics behavior is determined once the order is fixed. Furthermore, the numerical signatures are easily computable for the $\mathrm{fl}(\mathcal{N}(0,1))$ input distributions. This is because from the summation order we know the distributions $a \sim \mathrm{fl}(\mathcal{N}(0, \sigma_a^2))$ and $b \sim \mathrm{fl}(\mathcal{N}(0, \sigma_b^2))$ of each of the two addends of the $n-1$ floating-point additions. The second and fourth moment are thus modeled by $F(\sigma)\phi(r)$ and $G(\sigma)\psi(r)$ where $\sigma^2 = \sigma_a^2 + \sigma_b^2$ and $r = \min(\sigma_a^2/\sigma_b^2, \sigma_b^2/\sigma_a^2)$. Figure 5 shows the results.

V. Inner Product and Matrix Product

Consider an inner product "engine" for computing $\sum_{i=0}^{L-1} x_i y_i$ by computing a correctly rounded product $p_i = \mathrm{fl}(x_i y_i)$ followed by a SIMD sum as in Section IV on the vector $[p_0, p_1, \ldots, p_{L-1}]$. Once again, the variance of the error in the computed inner product can serve as signatures of the choice of SIMD length, for example, distinguishing between SIMD lengths of $\ell = 1, 2$, or 4. The main idea is to draw inputs in a controlled manner.

We used two input distributions. For the first input distribution, we begin with drawing one vector $\mathbf{x} = [x_0, x_1, \ldots, x_{L-1}]$ where each element is picked uniformly in $[1, 2]$, that is, $x_j \sim \mathrm{fl}(\mathcal{U}[1, 2])$. The vector $\mathbf{x}$ is then fixed. Then each inner product experiment consists of picking a different vector $\mathbf{y}$ where each element is $y_i = \mathrm{fl}(\xi_i)$, $\xi_i \sim \mathcal{N}(0,1)/x_i$. Thus, each computed product $p_i = \mathrm{fl}(x_i y_i)$ is well modeled by rounding a real number from $\mathcal{N}(0,1)$ since the unrounded products generally have twice the number of mantissa bits. The second and fourth moments of the rounding error are modeled by $F(1)$ and $G(1)$, respectively. The total error in this inner product computation consists of the $L-1$ error terms we tabulated for the SIMD summation process, with the addition of these L errors incurred by rounding $x_i y_i$ to $p_i = \mathrm{fl}(x_y y_i)$.

The second input distribution consists of first drawing one vector $\mathbf{x} = [x_0, x_1, \ldots, x_{L-1}]$ with $x_j \sim \mathrm{fl}(\mathcal{U}[1, 2])$ as before. Then each inner product experiment consists of picking different $\mathbf{y}$ vectors where every element is from $\mathrm{fl}(\mathcal{N}(0,1))$. The second and fourth moments for rounding the products $p_i = \mathrm{fl}(x_i y_i)$ are therefore modeled by $F(x_i)$ and $G(x_i)$, respectively. The analysis for summation with different SIMD lengths is easily generalized to where each summand $p_i \sim \mathrm{fl}(\mathcal{N}(0, x_i^2))$. The second and fourth moments of each of the $L-1$ error terms in the summation remain in the form $F(\sigma)\phi(r)$ and $G(\sigma)\psi(r)$ where the specific values of σ and r are fully determined by $\mathbf{x}$ alone.

Figure 6 shows the results of these two experiments with $L = 64$ and using $N = 10^4$ samples of inner products for each numerical variant. However, note that the error statistics for the second input distribution depend on the exact summation order.

Matrix computations $C_{m\times n} = \mathrm{fl}(X_{m\times k}\, Y_{k\times n})$ are merely $m \times n$ length-k inner products. Let $\Delta = C_{m\times n} - X_{m\times k} \times Y_{k\times n}$. If we set m to be 1, then the first distribution we used for inner products yields $\Delta_{1,j}^2$, $j = 1, 2, \ldots, n$ that are independent and identically distributed (iid). For a general m, the second input distributions lead to $\Delta_{i,j}^2$ where each row contains in it n iids. In particular, $\tilde{\Delta}_j \stackrel{\text{def}}{=} \sum_{i=1}^{m} \Delta_{i,j}^2$ for $j = 1, 2, \ldots, n$ are also iids. The signatures of each of these iids are obtainable in the same way described in this section.

VI. Fixed-Float Accumulator

Consider an accelerated summation unit via nonstandard arithmetic. In standard floating-point arithmetic, summing ℓ floating-point numbers requires $\ell - 1$ pairwise floating-point additions thus needing $\ell - 1$ alignments, that is, shifting of the operands' mantissas based on exponent differences. Moreover, the summation process is serial in nature as each alignment is based on the addends' exponent values, one of which is unknown until a previous summation is complete.

To accelerate the summing of ℓ floating-point values, one can employ a hybrid of fixed-point and floating-point arithmetic. Instead of aligning each addend with the latest partial sum, one can simply independently align each of the ℓ addends with the maximum exponent of these ℓ addends. We call this maximum exponent the alignment exponent. The mantissas with smaller exponents are right shifted and rounded. Optionally, we can also retain $d \geq 0$ extra bits beyond the least-significant-bit position of the alignment exponent. The exact sum of these ℓ aligned values is computed and subsequently rounded to the underlying floating-point format. Alignment with the largest exponents of an input argument is a floating-point type of arithmetic. But once this alignment exponent is obtained, the computation described here aligns multiple numbers with this "fixed" most-significant-bit position, and thus offering a type of fixed-point arithmetic. See Figure 7.

Since the numerics is well specified, the rounding errors are also well-defined stochastic processes for a fixed distribution of input values. We support this by computing the variance of a stochastic model of the rounding errors of this fixed-float accumulation of ℓ IEEE single-precision numbers $x_1, x_2, \ldots, x_\ell$, $x_i \sim \mathrm{fl}(\mathcal{N}(0,1))$. We show that the computed variances match the sampled variance experimentally.

We start with an analysis of the rounding errors. This fixed-float accumulation incurs two kinds of errors. First, each element x_i and up to $\ell - 1$ of them can incur a rounding error if its exponent is smaller than the alignment exponent by more than d bits. We call this the alignment error. After alignment, the exact sum of the aligned values must be converted back to an underlying floating-point format, thus possibly incurring one rounding error whenever the exponent of this exact sum

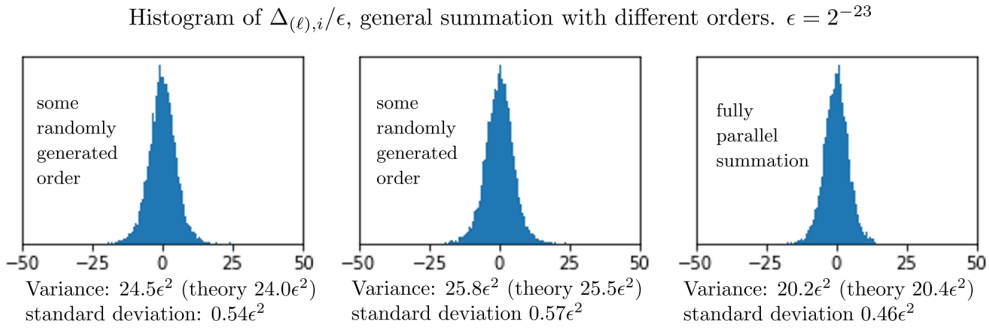

Fig. 5. Illustration of general sum. The first two figures correspond to two randomly generated order. The third order corresponds to a fully parallel sum.

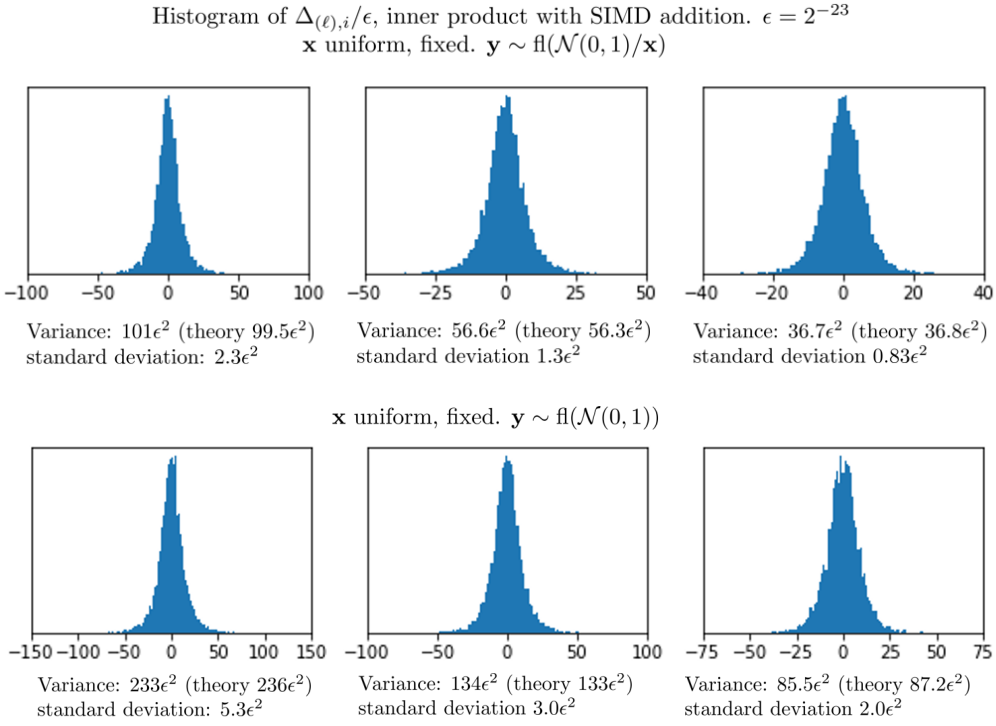

Fig. 6. Illustration of using observed mean square error as a signature of the underlying inner product numerics

exceeds $e_{\mathcal{A}} - d + 1$. We call this the conversion error. We model each of the two errors in turn.

A. Alignment Error

Let v_j be the expected square error of "rounding" j fractional bits to an integer. That is, $v_j = \frac{1}{8}, \frac{3}{32}, \frac{11}{128}, \ldots$ for $j = 1, 2, 3, \ldots$ and ultimately converging to $1/12$, assuming the round-to-nearest mode. Denote the random variable of alignment error of an addend by $\Delta_{\mathcal{A},\ell}$, then

$$E(\Delta^2_{\mathcal{A},\ell}) = 2^{-2(23+d)} \sum_{e_a} \sum_{j \geq 1} 2^{2e_a} p^{(\mathcal{A})}_{\mathrm{rnd}_j}(e_a)\, v_j, \qquad (11)$$

where e_a is the alignment exponent of ℓ floating-point addends and $p^{(\mathcal{A})}_{\mathrm{rnd}_j}(e_a)$ is the probability that the alignment exponent is e_a and that j bits need to be rounded off. This $p^{(\mathcal{A})}_{\mathrm{rnd}_j}(e_a)$ is determined in turn by two probabilities. The first is our familiar $p_k(1)$ (Equation 1), the probability of $|x| \in I_k$ where $x \sim \mathcal{N}(0,1)$. For simplicity, we use p_k instead of $p_k(1)$ throughout our discussion of the fixed-float accumulator. The second probability is $\hat{p}_k \stackrel{\text{def}}{=} \sum_{j<k} p_k$, the probability that the exponent of x is strictly less than k. From these, we determine $q_{k,m}$, the probability that the alignment exponent for m elements is k. This is the sum of the probabilities of all

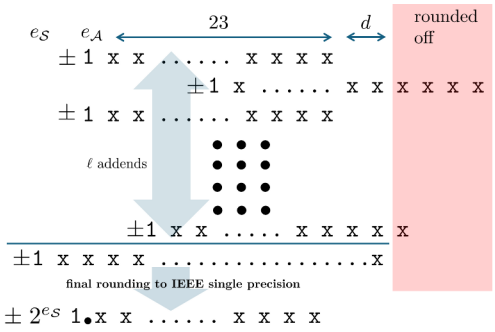

Fig. 7. Given ℓ input floating-point addends, the maximum exponent $e_{\mathcal{A}}$ is determined. Each addend is aligned to $e_{\mathcal{A}}$ and possibly rounded off beyond $23+d$ mantissa to the right of $e_{\mathcal{A}}$. These aligned addends are summed exactly, yielding a value of exponent $e_{\mathcal{S}}$. A final "conversion" to IEEE single precision is then performed. The errors incurred are (1) up to $\ell-1$ rounding errors during alignment, and (2) the final rounding due to conversion.

possible combinations of the m exponent values where some of the exponents are exactly k while the rest are strictly less. Thus

$$q_{k,m} = \sum_{i=1}^{m} \binom{m}{i} p_k^i \hat{p}_k^{m-i} = (p_k + \tilde{p}_k)^m - \hat{p}_k^m. \qquad (12)$$

We use Equations 1 and 12 to compute

$$p^{(\mathcal{A})}_{\mathrm{rnd}_j}(e_a) = q_{e_a,\ell-1}\, p_{e_a-d-j},$$

which then allow us to compute $E(\Delta^2_{\mathcal{A},\ell})$ using Equation 11. Moreover, $E(\Delta^4_{\mathcal{A},\ell})$ is a straightforward modification of Equation 11 as

$$E(\Delta^4_{\mathcal{A},\ell}) = 2^{-4(23+d)} \sum_{e_a} \sum_{j\geq 1} 2^{4e_a} p^{(\mathcal{A})}_{\mathrm{rnd}_j}(e_a)\, w_j \qquad (13)$$

where w_j is the 4-th moment of error when rounding j fractional bits to an integer. Equations 11 and 13 allow us to compute the value of $\mathrm{Var}(\Delta^2_{\mathcal{A},\ell})$ as well.

B. Conversion Error

The second error is that of rounding the exact sum of the aligned values to single precision. We compute the probability of needing to round j bits off the final sum. Let $e_{\mathcal{A}}$ be the alignment exponent and $e_{\mathcal{S}}$ be the exponent of the final sum. We model the number of bits to be rounded off as $j = e_{\mathcal{S}} - e_{\mathcal{A}} + d$ whenever $j \geq 1$. This is a slightly simplified model that assumes the least significant bit of the aligned sum is at $2^{e_{\mathcal{A}}-23-d}$. Since adding ℓ numbers cannot generate more than $\lceil \log_2(\ell) \rceil$ carry outs, we have $e_{\mathcal{S}} \leq e_{\mathcal{A}} + \lceil \log_2(\ell) \rceil$, and thus $j \leq d + \lceil \log_2(\ell) \rceil$.

$$E(\Delta^2_{\mathcal{S},\ell}) = 2^{-46} \sum_{e_a} 2^{2e_a} \sum_{j=1}^{\lceil \log_2(\ell) \rceil + d} \alpha_{e_a,j}, \quad \text{where}$$
$$\alpha_{e_a,j} = \mathrm{Prob}(e_{\mathcal{A}} = e_a \text{ and } e_{\mathcal{S}} = j + e_a - d)\, v_j$$

Similarly (cf. Equation 13)

$$E(\Delta^4_{\mathcal{S},\ell}) = 2^{-92} \sum_{e_a} 2^{4e_a} \sum_{j=1}^{\lceil \log_2(\ell) \rceil + d} \beta_{e_a,j} \quad \text{where}$$
$$\beta_{e_a,j} = \mathrm{Prob}(e_{\mathcal{A}} = e_a \text{ and } e_{\mathcal{S}} = j + e_a - d)\, w_j.$$

Moreover,

$$\mathrm{Prob}(e_{\mathcal{A}} = e_a \text{ and } e_{\mathcal{S}} = j + e_a - d) = \int_{I_{j+e_a-d}} g_{e_a}(t)\, dt,$$

$g_{e_a}(t)$ being the density of the aligned sum where the alignment exponent equals e_a. Recall that the probability of the alignment exponent of ℓ elements in Equation 12 is

$$q_{e_a,\ell} = \sum_{i=1}^{\ell} \binom{\ell}{i} p_{e_a}^i \hat{p}_{e_a}^{\ell-i},$$

we have the density $g_{e_a}(t)$ to be

$$g_{e_a}(t) = \sum_{i=1}^{\ell} \binom{\ell}{i} (\overbrace{\rho_{e_a} \star \cdots \star \rho_{e_a}}^{i-\text{times}})(\overbrace{\hat{\rho}_{e_a} \star \cdots \star \hat{\rho}_{e_a}}^{(\ell-i)-\text{times}})(t)$$

where $\rho_{e_a}(t)$ and $\hat{\rho}_{e_a}(t)$ are the $\mathcal{N}(0,1)$ Gaussian density function $\frac{1}{\sqrt{2\pi}} e^{-t^2/2}$ zeroed out on $\{t | \lfloor \log_2(|t|) \rfloor \neq e_a\}$ and $\{t | \lfloor \log_2(|t|) \rfloor \geq e_a\}$, respectively. Their convolutions and $\int_{I_k} g_{e_a}(t)$ for specific values of k can be computed by numerical convolutions and quadrature (such as using the trapezoidal rule). Figure 8 shows $(\rho_1 \star \rho_1 \star \hat{\rho}_1 \star \hat{\rho}_1)(t)$ which corresponds to $\ell = 4$, two addends' exponents equal 1 and the two other, strictly less.

C. Fixed-Float Accumulator

The total error Δ_{ff} incurred in a ℓ-length fixed-float accumulator is the sum of the ℓ alignment errors together with one conversion error. If α_i, $i = 1, 2, \ldots, \ell$ are the alignment errors and γ is the conversion error, then $\Delta_{\mathrm{ff}} = \gamma + \sum_i \alpha_i$. Hence Δ_{ff} is of the form

$$\Delta_{\mathrm{ff}} = \tau_1 + \tau_2 + \cdots + \tau_{\ell+1}$$

and $\mathrm{Var}(\Delta_{\mathrm{ff}})$ and $\mathrm{Var}(\Delta^2_{\mathrm{ff}})$ can be computed as in Equations 7 and 9. Experimentation is easy to set up. For each $\ell = 4, 8, 16$ and with $N = 10^4$ we draw $\ell \times N$ single-precision elements from $\mathrm{fl}(\mathcal{N}(0,1))$, thus obtaining $X_{\ell \times N}$ input values. We align each column of $X_{\ell \times N}$ with the number of extra bits d set to 2. Alignment errors are collected from the first row of the aligned values. The aligned values are summed in double precision (the sum is thus exact) and then rounded back to single precision, yielding the result of our fixed-float accumulator. The difference between this final result and the exact sum of the aligned values provide samples of conversion error; and the difference between the final result and the double-precision sum of the input values is the total error incurred by the fixed-float accumulator. Table I shows the experimental mean square (i.e. variances) alignment, conversion and total errors. We also tabulated the deviations given by (c.f. Equation 10)

$$\text{deviation} = \frac{|\text{observed variance} - \text{model variance}|}{\sqrt{\mathrm{Var}(\Delta^2_{\mathrm{ff}})/N}}.$$

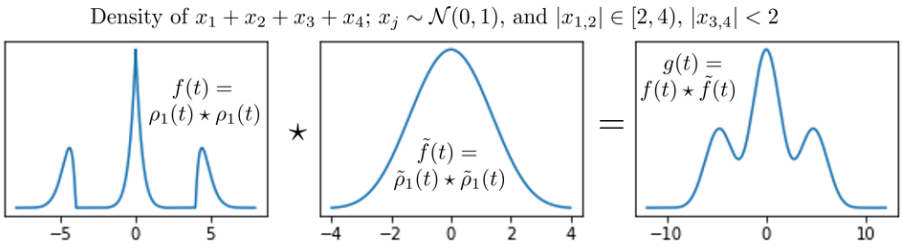

Fig. 8. This depicts the density function when 2 inputs have exponent of 1 and two have exponents strictly below 1.

	Mean Square Error (Variance): Sampled, Model, Deviation								
	$\ell = 4$			$\ell = 8$			$\ell = 16$		
Alignment	3.18e-17	3.01e-17	1.4	5.24e-17	5.31e-17	0.4	8.16e-17	8.30e-17	0.6
Conversion	2.68e-15	2.73e-15	0.7	5.27e-15	5.32e-15	0.4	1.04e-14	1.06e-14	0.5
Fixed-Float Accumulator	2.82e-15	2.85e-15	0.4	5.81e-15	5.75e-15	0.5	1.20e-14	1.19e-14	0.4

TABLE I

TABULATES FOR AN ℓ-LENGTH FIXED-FLOAT ACCUMULATOR THE OBSERVED VARIANCE, THEORETICAL VARIANCE, AND THEIR DEVIATIONS OF (1) ALIGNMENT ERROR OF ONE ADDEND, (1) THE CONVERSION ERROR, AND (3) THE TOTAL ERROR (WITH ℓ ALIGNMENT ERRORS)

VII. CONCLUDING REMARKS

This paper demonstrates through specific examples that the statistics of rounding errors can be used as signatures of the numerics that gives rise to them. These signatures can be used in a number of ways.

- *Error thresholds for numerical kernel testing*: One can use as signature the variance of error of a reference implementation whose quality we demand other implementations to meet. A threshold can be thus set at 3 standard deviations higher than the reference signature.
- *Hardware diagnostics:* In some situations, software developers need some reassurance on the numerical characteristics of some hardware components. If high-level description of the hardware defines the numerics well but not to the bitwise level, it is likely we can implement high-level references and obtain its rounding error signature. In this situation, matching signatures to high confidence level is the goal.
- *Designs exploration:* The rounding error signatures can be used as benchmarks of precisions when exploring different numerics designs especially when crucial kernels and common input distributions are identified.

We emphasize that we derive the statistics of various kernels here merely to demonstrate the reliability of these signatures. These signatures can in general be obtained through actual sampling on reference implementations of specific numerics and input distributions. Further studies can examine stochastic rounding (c.f. [16]) and low precisions arithmetic types such as 8-bit or 4-bit.

REFERENCES

[1] Nvidia, "NVIDIA A100 Tensor Core GPU architecture." [Online]. Available: https://images.nvidia.com/aem-dam/en-zz/Solutions/data-center/nvidia-ampere-architecture-whitepaper.pdf

[2] Graphcore, "Arithmetic for ai: A hardware perspective - mixed-precision arithmetic for ai: A hardware perspective." [Online]. Available: https://docs.graphcore.ai/projects/ai-float-white-paper/en/latest/index.html

[3] Tesla, "Tesla Dojo," Jun 2024. [Online]. Available: https://en.wikipedia.org/wiki/Tesla_Dojo

[4] M. Maddury, P. Kansal, and O. Wu, "Next gen mtia -recommendation inference accelerator," *2024 IEEE Hot Chips 36 Symposium (HCS)*, p. 1–27, Aug 2024.

[5] OCP, "OCP8-bit floating point specification (OFP8)." [Online]. Available: https://www.opencompute.org/documents/ocp-8-bit-floating-point-specification-ofp8-revision-1-0-2023-12-01-pdf-1

[6] ——, "OCP microscaling formats (MX) specification." [Online]. Available: https://www.opencompute.org/documents/ocp-microscaling-formats-mx-v1-0-spec-final-pdf

[7] M. Fasi, N. J. Higham, M. Mikaitis, and S. Pranesh, "Numerical behavior of nvidia tensor cores," *PeerJ Computer Science*, vol. 7, Feb 2021.

[8] J. H. Wilkinson, *Rounding errors in algebraic processes*. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2023.

[9] N. J. Higham, *Accuracy and stability of numerical algorithms*. SIAM, 2002.

[10] T. Ogita, S. M. Rump, and S. Oishi, "Accurate sum and dot product," *SIAM Journal on Scientific Computing*, vol. 26, no. 6, p. 1955–1988, Jan 2005.

[11] N. J. Higham and T. Mary, "A new approach to probabilistic rounding error analysis," *SIAM Journal on Scientific Computing*, vol. 41, no. 5, pp. A2815–A2835, 2019. [Online]. Available: https://doi.org/10.1137/18M1226312

[12] M. P. Connolly and N. J. Higham, "Probabilistic rounding error analysis of Householder QR factorization," *SIAM J. Matrix Anal. Appl.*, vol. 44, no. 3, pp. 1146–1163, 2023.

[13] D. Lohar, M. Prokop, and E. Darulova, "Sound probabilistic numerical error analysis," *Lecture Notes in Computer Science*, p. 322–340, 2019.

[14] E. Hallman and I. C. Ipsen, "Precision-aware deterministic and probabilistic error bounds for floating point summation," *Numerische Mathematik*, vol. 155, no. 1–2, p. 83–119, Aug 2023.

[15] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions*. Wiley, 1994.

[16] E.-M. E. Arar, M. Fasi, S.-I. Filip, and M. Mikaitis, "Probabilistic error analysis of limited-precision stochastic rounding," Aug 2024. [Online]. Available: https://arxiv.org/abs/2408.03069