EvalRound$^+$ Bootstrapping and its Rigorous Analysis for CKKS Scheme

Hyewon Sung[1], Sieun Seo[1], Taekyung Kim[2], and Chohong Min[1]

[1] Department of Mathematics, Ewha Womans University, South Korea
hyewonsung@ewha.ac.kr, sieun1114@ewha.ac.kr, chohong@ewha.ac.kr
[2] CryptoLab. Inc., South Korea
taekyung.kim@cryptolab.co.kr

Abstract. Bootstrapping stands as a fundamental component of fully homomorphic encryption (FHE) schemes, facilitating an infinite number of operations by recovering the ciphertext modulus. This work is aimed to significantly reduce the consumption of modulus in bootstrapping, thereby enhancing the efficiency of FHE performance, specifically for Cheon–Kim–Kim–Song (CKKS) scheme [8]. Building on EvalRound bootstrapping [14], which includes the steps of ModRaise, CoeffToSlot, EvalRound and SlotToCoeff, we introduce EvalRound$^+$ bootstrapping. This bootstrapping inherits the advantage of EvalRound bootstrapping in CoeffToSlot and resolves its disadvantage in SlotToCoeff. Furthermore, we conduct a set of rigorous and comprehensive analyses to precisely determine the optimal choices of the parameters. The implementation of EvalRound$^+$ bootstrapping, along with optimal choices, has achieved a reduction in modulus consumption by over 40% for CoeffToSlot and SlotToCoeff. Additionally, it has increased the number of levels for general multiplication by 2-4 in the most widely used bootstrapping parameter sets.

1 Introduction

The Cheon–Kim–Kim–Song (CKKS) scheme [8] is one of the fully homomorphic encryption (FHE) schemes enabling arithmetic computation over encrypted real or complex number data. Other FHE schemes also have their own special computing capabilities, but the CKKS scheme has been distinguished because of its adaptability for real world applications as it deals with real/complex numbers and vectors. The original CKKS scheme as described in [8] only deals with a finite number of multiplications; after each multiplication, the size of the noise in the encrypted message is doubled, and the subsequent rescaling process for removing such noise growth causes loss of certain amount of available modulus bits in the ciphertext.

To remedy this phenomenon, an algorithm called *(approximated) bootstrapping* has been proposed for refreshing any such "deteriorated" ciphertexts into nearly fresh ones, and one can keep doing further multiplication on them. Bootstrapping is an algorithm originally from Gentry's groundbreaking paper [9] that

opens up the new world of fully homomorphic encryption (FHE). In [6], the authors invented the algorithm of approximated bootstrapping to make the original CKKS scheme into a FHE scheme. Since then, many suggestions and new methods based on the approximated bootstrapping have appeared to improve its performance by several orders of magnitude. The adoption of the residue number system (RNS) [7], baby-step giant-step algorithm in the linear transformation steps in the bootstrapping [10], efficient use of gadget decomposition [12] and FFT-matrix grouping [5, 11] are some of noticeable achievements. Further improvements include optimal minimax polynomial approximation [15] and direct polynomial approximation for the modular reduction step for minimizing error variance [16], sine-series approximation [13], and Meta-BTS for enhancing bootstrapping precision by repeating the algorithm in a clever way [1].

Since the invention of the CKKS bootstrapping, its many major improvement techniques were well utilized and culminated in the work of Bossuat et al. [4], which is thus chosen as a reference algorithm to compare performances, and referred to as the *conventional* bootstrapping algorithm throughout this work. Especially, this conventional bootstrapping algorithm follows the well-received blueprint of the CKKS bootstrapping originated from [6], consisting of ModRaise, CoeffToSlot (CTS), EvalMod (EM) and SlotToCoeff (STC) in this very order.

Kim et al. [14] proposed EvalRound bootstrapping which is an addition of two shortcuts as outlined in Figure 3. The shortcuts connect four step of the conventional bootstrapping via subtractions, thus no significant additional computation cost is needed. The first subtraction leads to canceling the error of CTS, allowing for a significant reduction of modulus consumption in CTS. However, STC in the EvalRound algorithm is required to operate on qI, unlike STC of conventional operating on pt. Since $\|qI\|$ is much larger than $\|pt\|$, the error of STC of the EvalRound becomes as much larger than that of the conventional algorithm and consumes more modulus bits in STC. Therefore, depending on the parameter set, modulus consumption can be higher for EvalRound bootstrapping compared to the conventional one. In other words, EvalRound bootstrapping sometimes outperforms the conventional bootstrapping, while other times it performs worse. This difference is determined by the ratio of the size of the base prime and the scale factor Δ. For instance, parameter set P2 in [14] offers savings on modulus usage compared to the conventional one. On the other hand, with parameter set I in [4], it ends up consuming more modulus bits than the conventional bootstrapping, as illustrated in Figure 1.

We endeavored to enhance EvalRound bootstrapping for keeping its efficiency in CTS and removing its deficiency in STC. Our proposal also consists of two shortcuts. The first shortcut is exactly same as that of EvalRound, but the second one is different in its destination that is placed before STC, not after. Secondly, we did a thorough, rigorous mathematical error analysis for each step CTS, STC and EvalMod of the bootstrapping algorithm, elaborated upon in Section 4. As a result, we observe there is a certain threshold value for the size of the CTS error so that the total error starts to grow if the CTS error exceeds

the threshold. This allows us to fine-tune parameters under the theme of error balancing. Lastly, in Section 7, we introduce a novel discovery regarding the *sparsity patterns*, stemming from encoding FFT matrices. This discovery allows for the extra conservation of modulus bits in both CTS and STC.

1.1 Our Contribution

This work is aimed at reducing the consumption of modulus in bootstrapping and thus enhancing the efficiency of FHE performance. We make four contributions towards the aim. The first one is to introduce **EvalRound$^+$** bootstrapping that keeps the advantage of EvalRound bootstrapping in CTS and resolves its deficiency in STC. As shown in Figure 1, EvalRound$^+$ and EvalRound spend less modulus in CTS than the conventional bootstrapping, and EvalRound consumes very large modulus in STC, while EvalRound$^+$ and the conventional bootstrappings do not.

In both of EvalRound and EvalRound$^+$, the error size of CTS is allowed to be much larger than the conventional, because the error cancels out. The error size, however, can not be unboundedly large. There exists a certain **threshold value** beyond which the overall error starts to grow rapidly. To seek for a moderate size of the error, an empirical analysis with ad hoc parameter was introduced in [14]. Our second contribution is to build up a **rigorous and comprehensive analysis** to determine a precise estimate of the threshold value. Figure 4 shows that out estimate precisely hits the threshold value, while the earlier work does not. The difference in the estimates leads to the large difference in modulus consumption in CTS, as shown in Figure 1.

Our analysis decompose the overall error into three errors. In many of widely used bootstrapping parameter sets, we observed that the error of STC is noticeably smaller than the other two. The modulus consumption in the sets can be said to be unnecessarily excessive in STC. Our third contribution is to propose proper parameters that fix the imbalance in error sizes. The **error balancing** leads to a saving of large modulus in STC, depicted in Figure 1.

A homomorphic evaluation of linear transformation results in error whose size is, in general, linearly proportional the size of data. In the case of FFT matrices, we observed that the general estimation does not precisely hold. A thorough investigation on the observation revealed that there exists a special feature on the homomorphic evaluation of FFT matrix multiplications. Each diagonal vector of FFT matrices exhibits a **sparsity pattern**, when it is encoded. Taking the sparsity into consideration, the estimation, which is our fourth contribution, becomes precise. The widely used bootstrapping sets turn out to be somewhat excessive both in CTS and STC. The removal of the excessive consumption results in saving modulus in CTS and STC, as shown in Figure 1.

FHE developers usually choose optimal parameters empirically using exhaustive search over parameter space. Our error analysis and parameter design are based on rigorous mathematical theory, providing developers with a practical benefit by eleminating the need for the exhaustive search. The entire process of modulus reduction in CTS and STC, achieved through the implementation

of EvalRound$^+$, error balancing, and sparsity pattern applied to conventional bootstrapping, has been experimentally validated using our C language implementation. The code will be made **publicly accessible on GitHub**, allowing anyone to independently verify all the results presented in this paper.

Parameter	h	N	Δ	$\log(QP)$	L	$\log p_j$
I	192	2^{16}	2^{40}	1546	25	$5*61$
II	192	2^{16}	2^{30}	1552	25	$5*61$
III	192	2^{16}	2^{42}	1555	25	$3*59+2*60$

Parameter	$\log q_i$				
	q_0	Mult	STC	EvalMod	CTS
I	60	$9*40$	$3*39$	$8*60$	$4*56$
II	55	$5*60+5*30$	$2*45$	$8*55$	$4*53$
III	58	$9*42$	$3*42$	$9*58$	$3*58$

Table 1. Three conventional parameter sets I and II from [4] and III from [2]

1.2 Technical Overview

EvalRound$^+$ Bootstrapping : Our first contribution, that is referred to as EvalRound$^+$ bootstrapping, resolves the deficiency of EvalRound bootstrapping in STC, while it keeps the efficiency of EvalRound bootstrapping in CTS. Like its predecessor, EvalRound$^+$ consists of two skip connections. The first skip connection is exactly same in both bootstrappings, but the second one differs in its terminal position: one is before STC and the other is after. As illustrated in Figure 3, STC of EvalRound$^+$ operates on pt and that of EvalRound on qI. A homomorphic evaluation of linear transformations requires an amount of modulus that is proportional to the size of data. Since $\|pt\|$ is much smaller that $\|qI\|$, EvalRound$^+$ spends much less modulus in STC than EvalRound.

Threshold value : The work of EvalRound argued that the error in bootstrapping out is small, although the error of CTS is quite large. An experimental result in Figure 4 indicates that the total error is independent of the error of CTS, confirming the argument when the error is small. However the total error starts to grow beyond a certain threshold value. To seek for a moderate size of the error, an empirical analysis with ad hoc parameters was introduced in [14]. Our second contribution is an introduction of rigorous mathematical analysis that explains why the threshold exists and precisely estimates its value.

Error Balancing : Note that the total error is just a sum of $e_{\mathrm{CTS}}, e_{\mathrm{STC}}$ and e_{EM} in both of conventional and EvalRound$^+$ bootstrappings, as depicted in Figure 3. Significantly, imbalances among these errors are observed across all

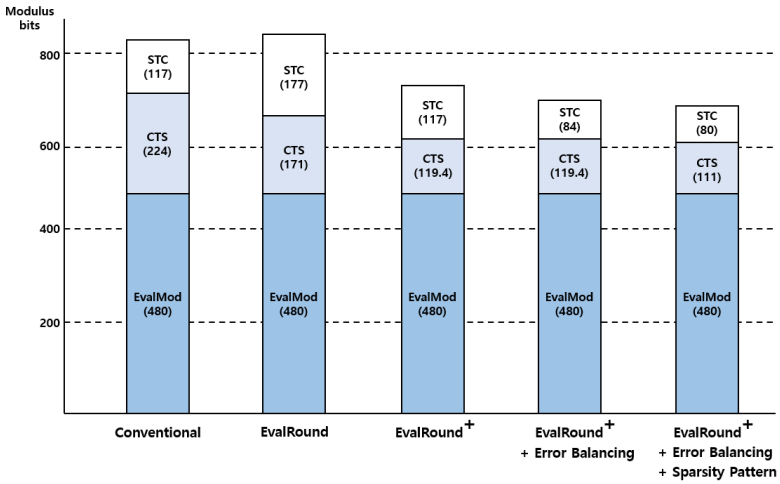

Fig. 1. Modulus consumption in CTS, STC and EvalMod for the conventional [4], EvalRound [14], and the three proposed bootstrappings under Parameter set I in Table 1. Considering the three proposed methods within the given parameter set results in a total savings of 150 bits compared to the conventional [4].

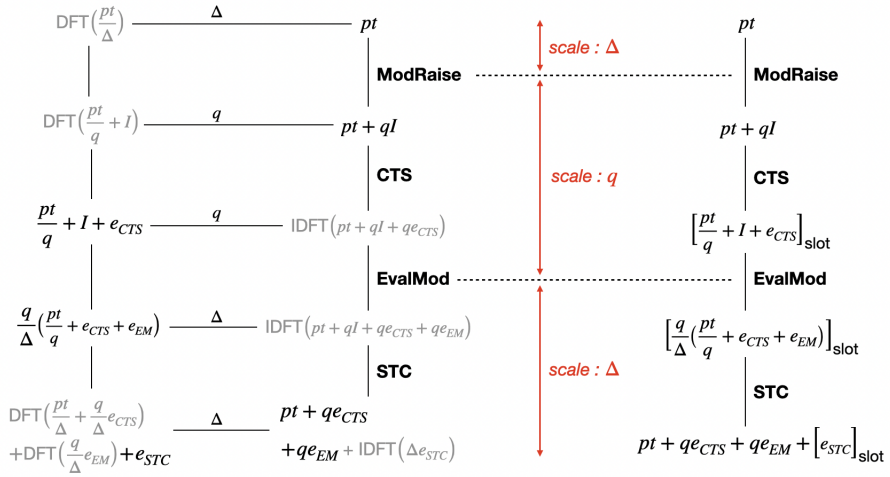

Fig. 2. Schematics of the conventional bootstrapping. The left one is a full diagram illustrating the state changes in both the slot side and coefficient side for each step. The plaintext pt emerges in one of the two sides during each step. The right one is a summary diagram tracking the side containing pt. We distinguish between the two sides by using $[\ ]_{\text{slot}}$ for the slot side. Consecutive states with the same encoding factor are merged and depicted as a unified region with up-down arrows.

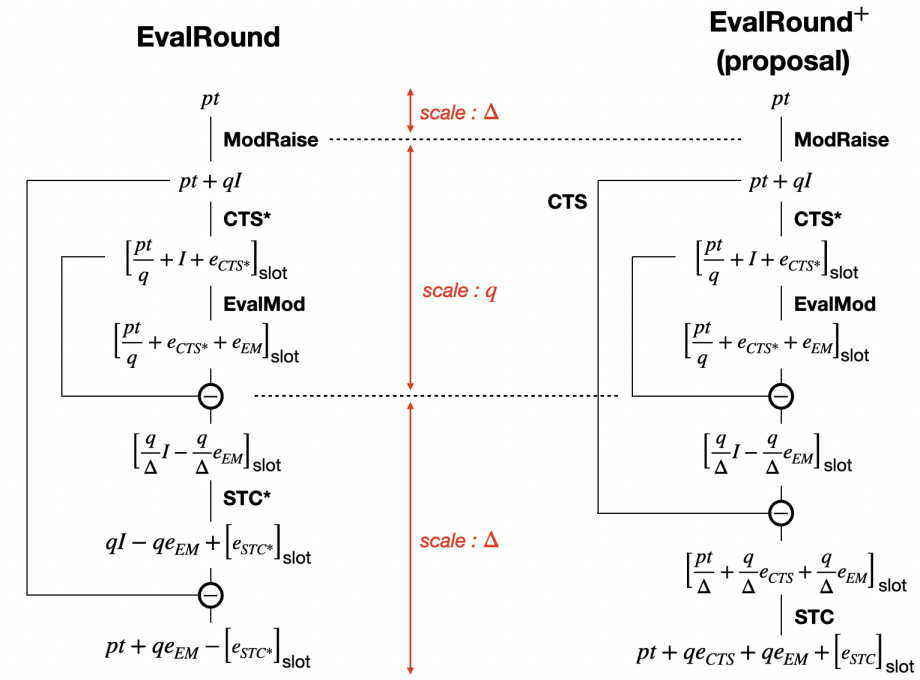

Fig. 3. Schematics of the EvalRound bootstrapping and the proposal. Each schematic is a summary diagram following the styles and notations outlined in Figure 2. Both bootstrappings add two skip connections to the conventional bootstrapping. The first skip connection leads to canceling e_{CTS^*} and saving the modulus consumption in CTS^* in both. However, the two bootstrappings differ in the terminal position of the second skip connection: it is after STC in EvalRound and before STC in the proposal.

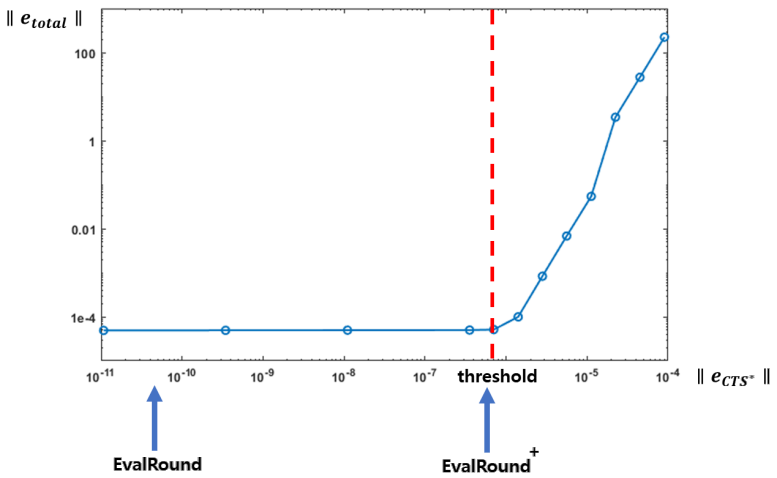

Fig. 4. The plot of the total error $\|e_{\text{total}}\|$ in bootstrapping with respect to $\|e_{\text{CTS}^*}\|$ under the setting of Parameter set I in Table 1. However, there exists a certain threshold value of $\|e_{\text{CTS}^*}\|$ beyond which $\|e_{\text{total}}\|$ starts to grow. The threshold is revealed by our analysis and the exact proper $\|e_{\text{CTS}^*}\|$ occur in EvalRound$^+$ while unnecessarily excessive small $\|e_{\text{CTS}^*}\|$ occurs in EvalRound. Our analysis uncovers this threshold, indicating that the exact and suitable $\|e_{\text{CTS}^*}\|$ occurs in EvalRound$^+$, while excessively small values of $\|e_{\text{CTS}^*}\|$ occur in EvalRound.

parameter sets, as illustrated in Figure 5. This observation highlights that the currently widely used parameters allocate unnecessarily excessive modulus in STC. In this regard, there is a potential for reducing the excessive allocation while keeping total error. Here arises the question about the size of the reduction. Retaining the total accuracy, e_{STC} is allowed to grow up to the level that is the maximum of the other errors, that is referred to as error balancing. Precisely, our error analysis determine the optimal parameter to reach the level. As a result, modulus consumption in STC is reduced by 27.6% and 52.1% from that of conventional and EvalRound bootstrappings, respectively.

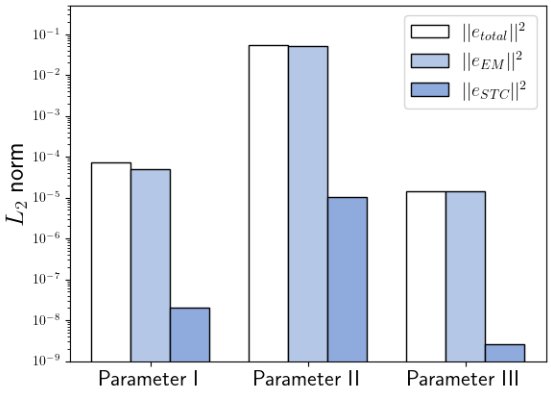

Fig. 5. Magnitude of e_{total}, e_{EM} and e_{STC} with Parameter I, II and III. Notably, e_{STC} is smaller to the other errors. This suggests that there is a potential for modular reduction of the scale factor in STC.

Sparsity Patterns : To homomorphically perform linear transformations, it is necessary to undergo the inverse fast Fourier Transform (iFFT) for the diagonal vectors of a matrix. When a diagonal vector of general matrices undergoes iFFT, the output vector is usually dense, not sparse. However, in the case of FFT matrices, the output vector of each diagonal vector is sparse. Note that the iFFT output of each column vector of FFT matrices is trivially sparse, but that of each diagonal is not.

Figure 6 shows two examples. One example is a general matrix and the other is a FFT matrix. Furthermore, the output vector of FFT matrices is not only sparse, but also periodic in the emergence of nonzero values, which we refer to as 'Sparsity Patterns'. The error of homomorphic linear transform is well-known to be inversely proportional to modulus consumption in general. However, our rigorous analysis reveals that the well-known estimation does not hold in the presence of sparsity pattern, leading to inordinate amount of modulus.

Our rigorous analysis introduces a new and acute estimation of error, taking into consideration of sparsity pattern. This enables us to reduce the inordinate amount of modulus to the optimal amount in both of CTS and STC.

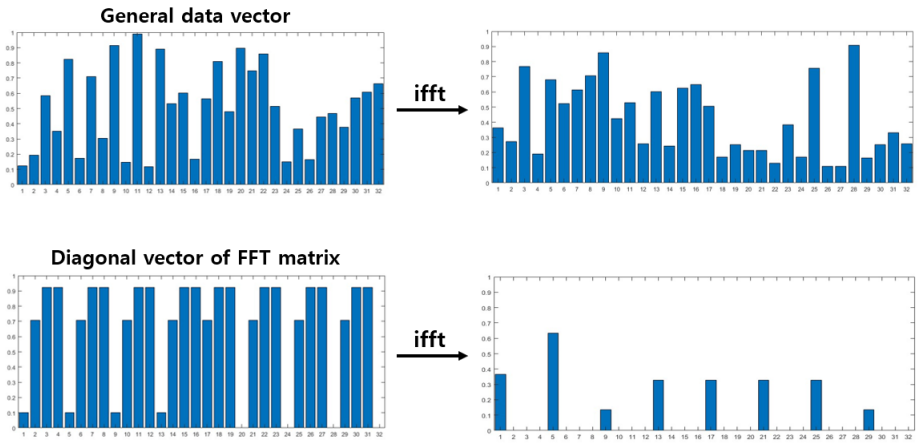

Fig. 6. This figure illustrates the phenomenon of the sparsity pattern. Specifically, it compares the results of applying iFFT to a general data vector and to a diagonal vector of the FFT matrix when $N = 32$. In the former case, there is no apparent sparsity pattern, while in the latter case, the existence of a sparsity pattern can be confirmed.

2 Preliminaries

2.1 CKKS homomorphic encryption scheme

We first recap the CKKS leveled homomorphic encryption scheme [8] in this section and the ones that follow. It will also serve us to fix notations for the rest of the paper.

Notation : For a power-of-two N, denote by $R = \mathbb{Z}[x]/(x^N + 1)$, the ring of integers of the $2N$-th cyclotomic field, which is a fundamental ring for the CKKS scheme and the RLWE problem the CKKS scheme is based on. For a positive q, let $R_q = R/qR = \mathbb{Z}_q[x]/(x^N + 1)$. Here N is determined at the parameter selection step of the CKKS scheme. A CKKS ciphertext can encrypt a complex vector of a power-of-two length which is maximally $N/2$. This vector is called a (complex) message. Here for the ease of description, we assume every message has an exact length of $N/2$. We denote by $\|\cdot\|$ L2 norm and when the input is a polynomial, this denotes the norm of coefficient vector.

Encoding and Decoding : Let ζ be a primitive $2N$-th root of unity contained in $\mathbb{C}$, e.g., $\zeta = \exp(\pi\sqrt{-1}/N)$, where $\sqrt{-1}$ is a complex imaginary unit. For integers i, write $\zeta_i := \zeta^{5^i}$. The map

$$\mathsf{DFT}_N : \mathbb{R}[x]/(x^N+1) \to \mathbb{C}^{N/2}, \quad m(x) \mapsto (m(\zeta_0), m(\zeta_1), \cdots, m(\zeta_{N/2-1})) \quad (1)$$

is known to be an isomorphism, with inverse iDFT_N. When the dimension N is understood, we also omit the subscript N so we write $\mathsf{DFT} = \mathsf{DFT}_N$ and $\mathsf{iDFT} = \mathsf{iDFT}_N$. With these algebraic maps, we can encode a complex message $z \in \mathbb{C}^{N/2}$ to a *plaintext* $\mathsf{pt} \in R$ and in reverse decode from pt to z.

- $\mathsf{Encode}(z; \Delta)$. For an $N/2$-dimensional vector z of complex numbers and a scale factor Δ, the encoding process first transforms z to a polynomial in $\mathbb{R}[x]/(x^N+1)$ and quantize it into an element of R. It returns

$$\mathsf{pt} = \mathsf{Encode}(z; \Delta) = \lfloor \Delta \cdot \mathsf{iDFT}(z) \rceil, \quad (2)$$

  where $\lfloor \cdot \rceil$ is the coefficient-wise rounding to the nearest integers.
- $\mathsf{Decode}(\mathsf{pt}; \Delta)$. For a plaintext pt and its scale factor Δ, the decoding process returns

$$z = \mathsf{Decode}(\mathsf{pt}; \Delta) = \mathsf{DFT}(\mathsf{pt}/\Delta). \quad (3)$$

  Here the polynomial pt/Δ is computed in $\mathbb{R}[x]/(x^N+1)$.

2.2 CKKS homomorphic encryption scheme

Bootstrapping in the CKKS scheme : A ciphertext ct is called of level ℓ if it has moduli $Q_\ell = q_0 \cdots q_\ell$. After a single multiplication of ct with another ciphertext or a plaintext, ct is at the same level ℓ but it has its internal scale factor effectively squared. In order to make the scale factor normal, we do the rescale operation, which brings ct's scale factor to a normal one but drops its level by one. After certain such multiplication and rescaling, ct reaches at the bottom level and we cannot do any further multiplication because it can cause decryption failure.

Bootstrapping can resolve this problem by recovering the ciphertext modulus, thereby allowing us to perform further homomorphic operations on the ciphertext. The conventional bootstrapping of Cheon et al. [6] consists of the following four steps: ModRaise, CoeffToSlot, EvalMod, and SlotToCoeff.

- ModRaise. It can be seen as the main operation of the bootstrapping algorithm conceptually, as it lifts the input ciphertext with near-bottom modulus q to one with the maximal modulus Q_L. As a side effect, it also alters the plaintext m encrypted in the ciphertext, by adding additional term qI with a polynomial with small coefficients I, resulting the encrypted plaintext to be $m + qI$. The sole purpose of the remaining stages of the bootstrapping algorithm is to remove this qI part from the plaintext.

– CoeffToSlot (CTS). Since the removal of qI, which is essentially a modular reduction operation(modular q), can only take place at the slot side, i.e. at the messages the ciphertext encrypts. Hence we need to transfer the plaintext $m + qI$ to the slot side, which is exactly what CTS is doing. The plaintext is an integer-valued vector and the transfer is to multiply the vector by the iDFT matrix. The entries of the matrix are multiplied by a so-called scale factor Δ_{CTS} and rounded to integers. The matrix multiplication is notoriously slow in computation, because it is a dense matrix. In practice, the matrix is decomposed into d_{CTS} number of sparse matrices and the matrix multiplication is replaced by fast successive multiplications by the sparse matrices.
– EvalMod. It is the modular reduction modular q operation. This can be achieved by evaluating a polynomial approximating the modular reduction function.
– SlotToCoeff (STC). It transfers the content in the slot side to the coefficient side, hence restoring the original plaintext. The transfer is the multiplication of DFT matrix. As in CTS, the matrix's entries are multiplied by a Δ_{STC} and rounded to integers, and the matrix is decomposed into d_{STC} number of sparse matrices.

Homomorphic Linear Transform : In CKKS bootstrapping, there are two major computations: linear transformation and polynomial evaluation. Since the part of CKKS bootstrapping that our paper aims to improve is implemented through linear transformations, we review linear transformation. Let $A \in \mathbb{C}^{\frac{N}{2} \times \frac{N}{2}}$ be a matrix with diagonals $v_1, \cdots, v_k$, and z be a data. Then, the result of multiplying A with z can be expressed as

$$Az = v_1 \odot z_1 + \cdots + v_k \odot z_k \,,$$

where each z_i is a rotation of $z \in \mathbb{C}^{\frac{N}{2}}$ with shift s_i. Assume that A and z are encoded into plaintexts by scale factor Δ. We denote the encoded forms of each v_i and z_i as pt_{v_i} and $\mathsf{rot}^{s_i}(pt_z)$, respectively. Then, a homomorphic evaluation of Az is given as below. See the details in [11].

$$pt_{v_1} * \mathsf{rot}^{s_1}(pt_z) + \cdots + pt_{v_k} * \mathsf{rot}^{s_k}(pt_z)$$

FFT decomposition in CTS and STC : CTS multiplies the iDFT matrix $\frac{2}{N}\overline{\mathrm{DFT}}^T$ to the message slots, in order to convert the polynomial coefficient representation into the slot representation. Since there are $\frac{N}{2}$ diagonals in iDFT matrix, it requires $\mathcal{O}(N)$ to multiply one single iDFT matrix homomorphically. To address this computational cost issue, Han et al. [12] proposed utilizing the FFT decomposition technique of the iDFT matrix. As a result of the decomposition, $\log N - 1$ number of FFT matrices are generated. In practice, considering the multiplicative depth, the multiplication is performed by grouping several FFT matrices together, and let us say d (e.g. $d = 3$ or 4), of matrices

$$\begin{aligned} \mathrm{DFT} &= E_1 E_2 \cdots E_{logN-1} \\ &= A_1 A_2 \cdots A_d. \end{aligned}$$

If d is assumed to divide the number of the FFT matrices for simplicity, each A_i is a product of $\frac{\log N-1}{d}$ number of FFT matrices. Since each FFT matrix has three diagonals and its matrix norm is $\sqrt{2}$, each A_i has up to $k = 2^{\frac{\log N-1}{d}}$ diagonals and matrix norm $\sqrt{2}^{\frac{\log N-1}{d}}$. Actually it holds that $\|A_i x\| = \sqrt{2}^{\frac{\log N-1}{d}} \|x\|$ for any x. Let $B_i = \overline{A_i}^T / \|A_i\|^2$ for each i, then we have a decomposition of the iDFT matrix into L^2-isometric matrices

$$\frac{2}{N}\overline{\mathrm{DFT}}^T = B_d \cdots B_2 B_1.$$

By directly applying the error analysis of linear transformation to the case of CTS, we can deduce the error of CTS. By adopting L^2-isometry relation, our error estimation of CTS expressed as equality rather than inequality expressions, allowing us to precisely analyze and estimate the error.

The STC step is exactly the inverse procedure of CTS. Here, we multiply the DFT matrix to the message slots, so to covert slot representation back into the polynomial's coefficient representation. In STC, there is also a computation cost issue based on the number of diagonal vectors in the multiplied matrices. To address this, the same decomposition technique used in CTS is applied to decompose a single DFT matrix into $\log N - 1$ FFT matrices. Similarly, in order to reduce the multiplicative depth in STC, the FFT matrices are grouped together, resulting in a total of d matrix multiplications as $\mathrm{DFT} = E_1 E_2 \cdots E_{logN-1} = A_1 A_2 \cdots A_d$. Since STC involves series of matrix multiplications, the error of STC can be derived by employing the error analysis of linear transformation. This error estimation can also give the exact error value rather than analyzing the range of errors using inequalities.

3 EvalRound$^+$: reducing $\Delta_{\mathbf{CTS}}$ and $\Delta_{\mathbf{STC^*}}$

Our main goal in this work is to significantly lessen the amount of modulus and levels spent in bootstrapping by reducing Δ_{CTS} and Δ_{STC}, while remaining the same precision. This section introduces a novel bootstrapping, referred to as EvalRound$^+$, that keeps the efficiency of EvalRound in CTS and resolves its deficiency in STC. EvalRound$^+$ consists of two skip connections, as described in the schematics in Figure 3.

A homomorphic linear transform on a data is required to convert its real/complex elements into integers. The conversion consists of scaling the elements by a large number, called scale factor and rounding them to their nearest integers. The error of the linear transform is inversely proportional to the scale factor and proportional to the size of the data. CTS is the homomorphic linear transform with iDFT matrix, and STC is that with DFT matrix. In the schematics, e_{CTS}, e_{EM} and e_{STC} denote the errors of CTS, EvalMod and STC, respectively. The total error of each bootstrapping is described in the schematics as follows.

$$\begin{aligned}
\text{Conventional} &: e_{\mathrm{total}} = q \cdot e_{\mathrm{CTS}} + q \cdot e_{\mathrm{EM}} + [e_{\mathrm{STC}}]_{\mathrm{slot}} \\
\text{EvalRound} &: e_{\mathrm{total}} = q \cdot e_{\mathrm{EM}} - [e_{\mathrm{STC}^*}]_{\mathrm{slot}} \\
\text{EvalRound}^+ &: e_{\mathrm{total}} = q \cdot e_{\mathrm{CTS}} + q \cdot e_{\mathrm{EM}} + [e_{\mathrm{STC}}]_{\mathrm{slot}}
\end{aligned}$$

In conventional bootstrapping, e_{CTS} is directly included in e_{total} . Thus, keeping the total precision severely restricts the scale factor of CTS, or the consumption of modulus in CTS. However, in EvalRound, the error of CTS is canceled out, owing to the first skip connection. This allows much larger magnitude of the error and enables the use of much smaller consumption of modulus, while keeping the total precision. Since the scale factors of two CTSs are quite different, CTS^* in the schematics denote the CTS with the economic consumption in EvalRound. $\mathrm{EvalRound}^+$ inherits EvalRound up to the first skip connection. The scale factors of CTS, that are indicators of modulus consumption, can be compared as follows.

$$\begin{aligned} \text{Conventional}: &\ \Delta_{\mathrm{CTS}} \\ \text{EvalRound}: &\ \Delta_{\mathrm{CTS}^*} \ll \Delta_{\mathrm{CTS}} \\ \mathrm{EvalRound}^+: &\ \Delta_{\mathrm{CTS}^*} \ll \Delta_{\mathrm{CTS}} \end{aligned}$$

EvalRound has the aforementioned advantage in CTS, but there exists price to pay for it. STC operates on $\mathrm{iDFT}(qI)$, instead of $\mathrm{iDFT}(\mathsf{pt})$. Since $\|qI\|$ is much larger than $\|\mathsf{pt}\|$ and iDFT is an isometry map, EvalRound should take a larger scale factor in STC than conventional bootstrapping, resulting in the deficiency in STC. $\mathrm{EvalRound}^+$ is devised to resolve this deficiency. It locates the terminal position of the second skip connection before STC, not after. As a consequence, STC now operates on $\mathrm{iDFT}(\mathsf{pt})$ to resolve the deficiency. The scale factors of STC are compared in the below.

$$\begin{aligned} \text{Conventional}: &\ \Delta_{\mathrm{STC}} \\ \text{EvalRound}: &\ \Delta_{\mathrm{STC}^*} \gg \Delta_{\mathrm{STC}} \\ \mathrm{EvalRound}^+: &\ \Delta_{\mathrm{STC}} \end{aligned}$$

In overall, $\mathrm{EvalRound}^+$ is as effective as EvalRound in CTS, and resolves its deficiency in STC. Its implementation does not need to build any new programmings, but just adding two connections to the existing programming calls is enough. In putting parameters in the implementation, Δ_{STC} is taken as same as in conventional and Δ_{CTS^*} is taken much smaller than Δ_{CTS}. Here comes a main question for the criteria of the smallness. The experiment in Figure 4 suggests that there is a threshold on the decrease of Δ_{CTS^*}. The error size commences to grow beyond the threshold, so that the optimal choice of Δ_{CTS^*} turns out to be the threshold. The cause of the threshold will be explained with details and its value will be estimated sharply in subsequent sections.

4 Rigorous Analysis of the threshold value

The first skip connection in both of EvalRound and $\mathrm{EvalRound}^*$ leads to the vanishing of large error e_{CTS^*} and opens a room to save much modulus while retaining the overall accuracy. However, there exists a certain threshold value of $\|e_{\mathrm{CTS}^*}\|$ beyond which the overall accuracy is not retained any more. Kim et al.

[14] introduced a criteria estimating the threshold as follows.

$$\Delta_{\mathrm{CTS}^*} \simeq \frac{1}{\frac{\epsilon\sqrt{N}}{C_2} - 1} \cdot \frac{C_1 \cdot N^{1+\frac{1}{2d}} \cdot q}{\Delta}$$

The criteria is based on an empirical estimate with ad hoc parameters C_1, C_2 and ϵ. The error of successive matrix multiplications was roughly estimated through a series of inequality. This section aims at presenting the precise estimate of the threshold value, not an empirical one through a set of rigorous and comprehensive analyses.

4.1 Causality Tracing

The total error consists of three errors that are e_{CTS}, e_{EM} and e_{STC}, as shown in the schematics of Figure 3. The error of a linear transform is determined solely by the size of data and the scale factor associated with the transformation. STC acts on data $\frac{pt}{q}$ with scale factor Δ_{STC}, and CTS on $pt + qI$ with Δ_{CTS}. Thus, e_{STC} and e_{CTS} are independent of e_{CTS^*}, having no relation. Since the total error is a sum of e_{CTS}, e_{STC} and e_{EM}, the link between the total error and e_{CTS^*} should be involved in e_{EM}. Thus, we can narrow down the causality from e_{CTS^*} to e_{total} as follows.

$$\text{Causality} : e_{\mathrm{CTS}^*} \Longrightarrow e_{\mathrm{EM}} \Longrightarrow \text{threshold}$$

Let $\mathrm{EvalMod}(t)$ refer the homomorphic evaluation of polynomial $p(t)$ that approximates the modular function $[t]_1$ with period one. The discontinuity of $[t]_1$ hinders the approximation from being accurate. Using the fact that t in bootstrapping is close to its nearest integer, the hindrance is circumvented by the use of smooth analytic function $\frac{\sin(2\pi t)}{2\pi} \simeq [t]_1$. Thus, there exists the following series of approximations from $\mathrm{EvalMod}(t)$ to $[t]_1$.

$$\mathrm{EvalMod}(t) \simeq p(t) \simeq \frac{\sin(2\pi t)}{2\pi} \simeq [t]_1 .$$

The error of the first approximation results from the homomorphic evaluation of the polynomial, and is denoted by e_{Poly}. The polynomial $p(t)$ is the minmax approximation of $\frac{\sin(2\pi t)}{2\pi}$ calculated by the Remez algorithm [17], and denoted by e_{Remez}. The sine function accurately approximates $[t]_1$ for each integer, so that the error of the third approximation is given from the Taylor series

$$\frac{\sin(2\pi t)}{2\pi} = [t]_1 - \frac{(2\pi)^2}{6}[t]_1^3 + \frac{(2\pi)^4}{120}[t]_1^5 - \cdots .$$

Since it is an alternating series, the first nonzero term becomes the dominant error of the Taylor series, denoted by e_{Taylor}. Thus, $e_{\mathrm{EM}} = \mathrm{EvalMod}(t) - [t]_1$ is decomposed as

$$e_{\mathrm{EM}} = e_{\mathrm{Poly}} + e_{\mathrm{Remez}} + e_{\mathrm{Taylor}}. \tag{4}$$

The input variable t is given in the schematics of Figure 3. Since I is an integer variable and $[t]_1$ is the modular reduction with period one, e_{Taylor} is formulated as below.

$$\begin{aligned} t &= I + \tfrac{pt}{q} + e_{\text{CTS}^*} \\ e_{\text{Taylor}} = \tfrac{(2\pi)^2}{6}[t]_1^3 &= \tfrac{(2\pi)^2}{6}\left(\tfrac{pt}{q} + e_{\text{CTS}^*}\right)^3 \end{aligned}$$

We pointed out the causality from e_{CTS^*} to $e_{\text{EM}} = e_{\text{Poly}} + e_{\text{Remez}} + e_{\text{Taylor}}$. Now, let us take a closer investigation on each decomposition of e_{EM}. The magnitude of error in homomorphic multiplications is proportional to the size of data. Hence $\|e_{\text{Poly}}\|$ is primarily determined by $\|t\| \simeq \|I\|$. In the usual setting, the size of I dominates those of $\frac{\mathsf{pt}}{q}$ and e_{CTS^*}, unless Δ_{CTS^*} is unreasonably small. Thus, e_{Poly} can be removed in the causality tracing. On the other hand, e_{Remez} arises from the minmax polynomial approximation of the sine function, having no relation with e_{CTS^*}. From these reasons, the causality between the threshold and e_{CTS^*} lies on just e_{Taylor} and nothing else.

$$\text{Causality} : e_{\text{CTS}^*} \Longrightarrow e_{\text{Taylor}} \Longrightarrow \text{threshold}$$

4.2 Estimating $\|e_{\text{Taylor}}\|^2$ through $\|e_{\text{CTS}^*}\|^2$

We traced the causality of the threshold and reached at $e_{\text{Taylor}} = \frac{(2\pi)^2}{6}\left[\frac{\mathsf{pt}}{q} + e_{\text{CTS}^*}\right]^3$ whose expansion is a weighted sum of momemts of the random variable e_{CTS^*}. To begin with, we introduce a precise L^2 estimate of $\|e_{\text{CTS}^*}\|^2$ and the statistical features of e_{CTS^*}.

Theorem 1. *(Error of a homomorphic linear transformation) Let $Homo(Az)$ denote the homomorphic evaluation of Az and $e = Homo(Az) - Az$ be its error. Then we have*

$$\mathbb{E}\left[\|e\|^2\right] = \frac{kN}{12\Delta^2}\|z\|^2,$$

where k is the number of diagonal vectors of $A \in \mathbb{C}^{\frac{N}{2}\times\frac{N}{2}}$ and Δ is the encoding scale factor of the diagonal vectors. Furthermore, each e_i follows the normal distribution with mean zero and variance $\leq \frac{kN}{12\Delta^2}\|z\|^2$.

Proof. given in Appendix.

Theorem 2. *(a series of conformal matrices) Let $Homo^{seq}(A_d \cdots A_1 z)$ be the sequential homomorphic evaluation of $A_d \cdots A_1 z$, where $A_1, \cdots, A_d$ are conformal matrices, and let $k_1, \cdots, k_d$ be the number of the diagonals of $A_1, \cdots, A_d$, respectively. Then the error $e = Homo^{seq}(A_d \cdots A_1 z) - A_d \cdots A_1 z$ satisfies*

$$\mathbb{E}\left[\|e\|^2\right] = \frac{N\|z\|^2}{12\Delta^2}\|A_d\|^2 \cdots \|A_1\|^2 \left(\frac{k_d}{\|A_d\|^2} + \cdots + \frac{k_1}{\|A_1\|^2}\right).$$

Furthermore, each coordinate e_i follows a normal distribution with mean zero.

Proof. given in Appendix.

Let CTS* be conducted by a series of matrix multiplications that are from a FFT decomposition DFT $= A_d \cdots A_1$ with scale factor Δ_{CTS^*}. Each FFT block is conformal and the above general theorem estimates $\|e_{\mathrm{CTS}^*}\|^2$. Furthermore, it will be shown in later section that there exists a special feature with FFT matrices, so called 'Sparsity pattern' to refine the inaccurate estimate to be the precise one. For each i, S_i denotes the sparsity gap of the matrix A_i, which will be explained in details later.

$$\mathbb{E}\left[\|e_{\mathrm{CTS}^*}\|^2\right] \simeq \frac{N}{12\left(\Delta_{\mathrm{CTS}^*}\right)^2}\|A_d\|^2\cdots\|A_1\|^2\left(\frac{k_d}{\|A_d\|^2}+\cdots+\frac{k_1}{\|A_1\|^2}\right)\left\|\mathrm{DFT}\left(I+\frac{\mathsf{pt}}{q}\right)\right\|^2$$

$$\mathbb{E}\left[\|e_{\mathrm{CTS}^*}\|^2\right] = \frac{N}{12\left(\Delta_{\mathrm{CTS}^*}\right)^2}\|A_d\|^2\cdots\|A_1\|^2\left(\frac{k_d}{\|A_d\|^2 S_d}+\cdots+\frac{k_1}{\|A_1\|^2 S_1}\right)\left\|\mathrm{DFT}\left(I+\frac{\mathsf{pt}}{q}\right)\right\|^2$$

$$\|e_{\mathrm{CTS}^*}\|^2 \simeq \frac{N}{12\left(\Delta_{\mathrm{CTS}^*}\right)^2}\|A_d\|^2\cdots\|A_1\|^2\left(\frac{k_d}{\|A_d\|^2 S_d}+\cdots+\frac{k_1}{\|A_1\|^2 S_1}\right)\frac{h+1}{24}N^2$$

Here, we used the estimate $\left\|I+\frac{pt}{q}\right\|^2 = \frac{h+1}{12}N$ in [4] and the fact that the variance of random variable $\|e_{\mathrm{CTS}^*}\|^2$ becomes barely noticeable as N increases, e.g. 2^{16}. Thus we may omit the expectation symbol whenever dealing with macro quantities such as $\|e_{\mathrm{CTS}^*}\|^2$.

Now, we intend to estimate $\|e_{\mathrm{Taylor}}\|^2$ through $\|e_{\mathrm{CTS}^*}\|^2$. Utilizing the binomial expansion on $e_{\mathrm{Taylor}} = \frac{(2\pi)^2}{6}\left[\frac{\mathsf{pt}}{q}+e_{\mathrm{CTS}^*}\right]^3$, we obtain

$$\begin{aligned}
\mathbb{E}\left[\left(e_{\mathrm{Taylor}}[i]\right)^2\right] &= \mathbb{E}\left[\frac{(2\pi)^4}{36}\sum_{m=0}^{6}\binom{6}{m}\left(\frac{\mathsf{pt}[i]}{q}\right)^{6-m}\left(e_{\mathrm{CTS}^*}[i]\right)^m\right] \\
&= \frac{(2\pi)^4}{36}\sum_{m=0}^{6}\binom{6}{m}\left(\frac{\mathsf{pt}[i]}{q}\right)^{6-m}\mathbb{E}\left[\left(e_{\mathrm{CTS}^*}[i]\right)^m\right] \\
&= \frac{(2\pi)^4}{36}\sum_{m=0}^{3}\binom{6}{2m}\left(\frac{\mathsf{pt}[i]}{q}\right)^{6-2m}\mathbb{E}\left[\left(e_{\mathrm{CTS}^*}[i]\right)^{2m}\right].
\end{aligned}$$

According to Theorem 2, each $e_{\mathrm{CTS}^*}[i]$ follows a normal distribution with mean zero. In the above, we used the fact that every odd moment of the normal distribution is zero. The even moment is readily given [18] by its variance that is denoted by σ_i^2 to obtain

$$\mathbb{E}\left[\left(e_{\text{Taylor}}[i]\right)^2\right] = \frac{(2\pi)^4}{36}\left[\left(\frac{\mathsf{pt}[i]}{q}\right)^6 + \sum_{m=1}^{3}\binom{6}{2m}\left(\frac{\mathsf{pt}[i]}{q}\right)^{6-2m}\sigma_i^{2m}(2m-1)!!\right]$$
$$\leq \frac{(2\pi)^4}{36}\left[\left(\frac{\Delta}{q}\right)^6 + \sum_{m=1}^{3}\binom{6}{2m}\left(\frac{\Delta}{q}\right)^{6-2m}\sigma_i^{2m}(2m-1)!!\right]$$
$$\mathbb{E}\left[\left\|e_{\text{Taylor}}\right\|^2\right] \leq \frac{(2\pi)^4}{36}\left[\frac{N}{2}\left(\frac{\Delta}{q}\right)^6 + \sum_{m=1}^{3}\binom{6}{2m}\left(\frac{\Delta}{q}\right)^{6-2m}(2m-1)!!\left(\sum_{i=1}^{N/2}\sigma_i^{2m}\right)\right]$$
$$= \sum_{m=1}^{3}\left(\sum_{i=1}^{N/2}\sigma_i^{2m}\right)\left[\frac{(2\pi)^4}{36}\binom{6}{2m}\left(\frac{\Delta}{q}\right)^{6-2m}(2m-1)!!\right] + \frac{(2\pi)^4}{36}\frac{N}{2}\left(\frac{\Delta}{q}\right)^6.$$

Here, we use the usual assumption $|\mathsf{pt}[i]| \leq \Delta$. The estimation of each individual σ_i is cumbersome, but the sum of their squares equals $\left\|e_{\text{CTS}^*}\right\|^2$. Based on empirical observations, we note that most of σ_i^2 lie between a half of their average $\frac{2}{N}\left\|e_{\text{CTS}^*}\right\|^2$ and one and a half , and postulate the following estimation.

$$\sum_{i=1}^{N/2}\sigma_i^{2m} \leq \left(\frac{3}{2}\right)^m\left(\frac{2}{N}\left\|e_{\text{CTS}^*}\right\|^2\right)^m$$

Combining the postulate with the precise estimation of $\left\|e_{\text{CTS}^*}\right\|^2$, the overall estimation of $\left\|e_{\text{Taylor}}\right\|^2$ is given as

$$\left\|e_{\text{Taylor}}\right\|^2 \leq \sum_{m=1}^{3}\left(\sum_{i=1}^{N/2}\sigma_i^{2m}\right)\left[\binom{6}{2m}\frac{(2\pi)^4}{36}\left(\frac{\Delta}{q}\right)^{6-2m}(2m-1)!!\right] + \frac{(2\pi)^4}{36}\frac{N}{2}\left(\frac{\Delta}{q}\right)^6$$
$$\leq \sum_{m=1}^{3}\left(\frac{3}{2}\right)^m\left(\frac{2}{N}\left\|e_{\text{CTS}^*}\right\|^2\right)^m\left[\binom{6}{2m}\frac{(2\pi)^4}{36}\left(\frac{\Delta}{q}\right)^{6-2m}(2m-1)!!\right]$$
$$+ \frac{(2\pi)^4}{36}\frac{N}{2}\left(\frac{\Delta}{q}\right)^6.$$

4.3 Putting all together : optimal Δ_{CTS^*}

From the previous subsection, we analyzed e_{Taylor} in order to find its dependency on Δ_{CTS^*}. Continuing from here, in this section, we will utilize all the previous analyses and identify the relationship between e_{EM} and Δ_{CTS^*} to accurately determine the optimal Δ_{CTS^*}, the threshold value.

Taking into consideration the overall estimation of $\left\|e_{\text{Taylor}}\right\|^2$, Δ_{CTS^*} only influences on e_{CTS^*}, and the magnitude of $\left\|e_{\text{CTS}^*}\right\|^2$ is proportional to $\frac{1}{\Delta_{\text{CTS}^*}^2}$.

By expanding the estimation with respect to Δ_{CTS^*}, we can yield the following inequality for some $c_0, c_1, c_2, c_3 \in \mathbb{R}^+$.

$$\|e_{\mathrm{Taylor}}\|^2 \leq c_0 + \frac{c_1}{(\Delta_{\mathrm{CTS}^*})^2} + \frac{c_2}{(\Delta_{\mathrm{CTS}^*})^4} + \frac{c_3}{(\Delta_{\mathrm{CTS}^*})^6} \tag{5}$$

In the decomposition (4), the three errors in the right-hand-side differ from one another when and where they arise, and can be assumed to be independent to each other. Thus, their variances are additive and we have the following formulation.

$$\|e_{\mathrm{EM}}(\Delta_{\mathrm{CTS}^*})\|^2 \leq \|e_{\mathrm{Poly}}\|^2 + \|e_{\mathrm{Remez}}\|^2 + c_0 + \frac{c_1}{(\Delta_{\mathrm{CTS}^*})^2} + \frac{c_2}{(\Delta_{\mathrm{CTS}^*})^4} + \frac{c_3}{(\Delta_{\mathrm{CTS}^*})^6}$$

Note that the last three terms are monotonically increasing as Δ_{CTS^*} is increasing. They are unnoticeable small when Δ_{CTS^*} is large, but becomes noticeable when their sum is equivalent to the magnitudes of the first three terms. From these reasons, the threshold for Δ_{CTS^*} can be determined by solving the following algebraic equation for Δ_{CTS^*}.

$$\mathrm{Max}\left(\|e_{\mathrm{Poly}}\|^2, \|e_{\mathrm{Remez}}\|^2, c_0\right) = \frac{c_1}{(\Delta_{\mathrm{CTS}^*})^2} + \frac{c_2}{(\Delta_{\mathrm{CTS}^*})^4} + \frac{c_3}{(\Delta_{\mathrm{CTS}^*})^6} \tag{6}$$

The above equation has a unique solution, since the right hand side is a monotonically decreasing function. Moreover, it can be easily solvable by many efficient root finding algorithms such as bijection method.

Example 1. This example compares the difference between the CTS error estimations obtained using the EvalRound method [14] and our error estimation derived from Theorem 2, and demonstrates that our estimation yields much more accurate results. Additionally, this example ultimately shows that utilizing our estimation to determine Δ_{CTS^*} results in saving approximately twice as many modulus bits in CTS compared to the method proposed by EvalRound.

The table below compares the observed CTS errors for each parameter set from Table 1 with the CTS error estimations from EvalRound and our error estimation derived from Theorem 2. The comparison clearly shows that our estimation formula yields significantly more accurate predictions than the formula [14]. When considering the sparsity pattern in our estimation, it enables even more accurate predictions and a detailed explanation of this will be provided in Section 7.

	observation	EvalRound	our estimation by Theorem 2	our estimation with sparsity pattern
I	5.26×10^{-15}	2.85×10^{-14}	1.30×10^{-14}	5.99×10^{-15}
II	4.19×10^{-14}	7.48×10^{-9}	1.04×10^{-13}	4.26×10^{-14}
III	3.52×10^{-15}	1.36×10^{-16}	2.80×10^{-14}	6.78×10^{-15}

The table below compares the precise Δ_{CTS^*} value determined by EvalRound with it derived through our rigorous analysis for each parameter set. Comparing

the modulus bits used in CTS, EvalRound$^+$ allows for approximately half the number of modulus bits to be saved compared to the conventional bootstrapping method.

$\log(\Delta_{\mathrm{CTS}^*})$	conventional	EvalRound	EvalRound$^+$
I	56	42.75	29.85
II	53	48	27.98
III	58	39	30.51

5 Error Balancing: optimizing Δ_{STC}

The total error is a sum of three errors, as shown in the schematics of Figure 3. The error $[e_{\mathrm{STC}}]_{\mathrm{slot}}$ was observed to be notably smaller than the others in the widely used parameter sets. This observation implies that it is allowed to increase its bound while retaining the overall accuracy. The error bound is decided by Δ_{STC} through the estimation given in Theorem 2.

CTS performed in the beginning and affects the order routines in the sequential process of bootstrapping. From this reason, Δ_{CTS} not only affects e_{CTS} but also e_{EM}, as discussed in Section 4. However, STC is performed in the last and Δ_{STC} affects only e_{STC}. Thus, the smallest Δ_{STC} that maintains the overall accuracy can be calculated by the following relation.

$$\left\|[e_{\mathrm{STC}}]_{\mathrm{slot}}\right\|^2 = \frac{N}{2} \cdot \left\|e_{\mathrm{STC}}\left(\Delta_{\mathrm{STC}}\right)\right\|^2 = \max\left(\left\|e_{\mathrm{CTS}}\right\|^2, \left\|e_{\mathrm{EM}}\right\|^2\right)$$

Example 2. Figure 5 shows that the error of STC is much smaller than the other errors in all the parameter sets. Δ_{STC} is reduced by the above relation up to the limit that keeps the overall accuracy. The table below reports the exact value of Δ_{STC} in EvalRound$^+$ with the error balancing on and off.

	before error balancing		after error balancing	
	$\log(\Delta_{\mathrm{STC}})$	$\|e_{\mathrm{total}}\|$	$\log(\Delta_{\mathrm{STC}})$	$\|e_{\mathrm{total}}\|$
I	39	7.16×10^{-5}	28.3	6.32×10^{-5}
II	45	5.36×10^{-2}	25.7	5.76×10^{-2}
III	42	1.41×10^{-5}	28.0	1.40×10^{-5}

When comparing the modulus consumption of STC before and after error balancing, it is observed that there is a reduction of 32 bits, 58 bits and 42 bits for Parameter I,II and III, respectively. Through this result, we can verify that error balancing allows for a reduction in the excessive consumption of modulus bits in STC while maintaining the overall accuracy, and furthermore enables to choose an optimal Δ_{STC}.

6 Efficient implementation of the additional CTS

EvalRound$^+$ offers advantages in saving modulus and levels, yet it requires an additional iteration of CTS. As depicted in Figure 7, the additional CTS is

configured in parallel with the CTS*+EvalMod. Therefore, it can be executed in parallel within the computation time of CTS*+EvalMod. However, if sequential execution is preferred, it would incur some overhead. Fortunately, this overhead is not significant due to our design for the efficient implementation of the additional CTS. The parallelism facilitates the execution of the additional CTS within the extensive levels designated for CTS* and EvalMod. In this section, we detail the implementation of the additional CTS to achieve the fast implementation of EvalRound$^+$ by leveraging these reserved levels.

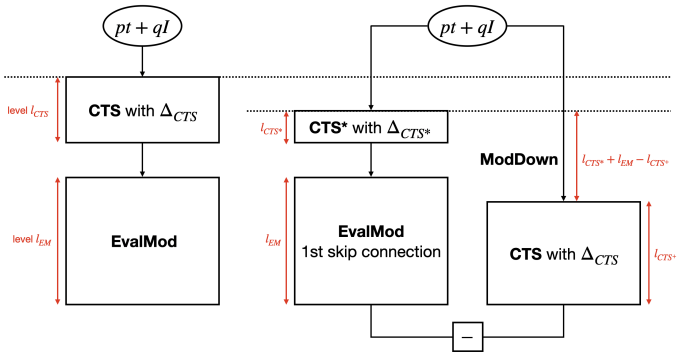

Fig. 7. Level consumption in the conventional bootstrapping(left) and the proposed EvalRound$^+$ (right). The two processes of EvalRound$^+$ can be executed independently and thus in parallel. Furthermore, CTS can utilize much larger levels $l_{\mathrm{CTS}^*} + l_{\mathrm{EM}}$ than l_{CTS} and l_{CTS^*} to minimize its calculation time. As a result, an optimal level l_{CTS^+} is decided in Section 6 and the remaining levels $l_{\mathrm{CTS}^*} + l_{\mathrm{EM}} - l_{\mathrm{CTS}^+}$ are ModDowned for its further speed-up.

For the sake of clarity in subsequent discussions, we will refer to this additional CTS as CTS$^+$ exclusively in this section. Since CTS$^+$ is essentially equal to CTS, it is also represented through the product of FFT matrices as [11]. However, in contrast to CTS, levels $l_{\mathrm{CTS}^*} + l_{\mathrm{EM}}$ are given for CTS$^+$, where l_{CTS^*} and l_{EM} are the levels for CTS* and EvalMod, respectively. Thus the number of matrix groups in CTS$^+$ significantly exceeds that in CTS*, offering greater flexibility in grouping the FFT matrices. We note that, as represented in [11], each FFT matrix E_i has only three diagonal vectors and each r consecutive multiplication of FFT matrices has at most $2r-1$ diagonal vectors. Therefore, they discussed a trade-off between the depth and the number of diagonal vectors. However, the design of CTS$^+$ possesses substantial number of available levels $l_{\mathrm{CTS}^*} + l_{\mathrm{EM}}$. Thus, we will opt for a strategy that emphasizes reducing the total number of diagonal vectors over decreasing depth.

Counting the number of diagonals in $[11, 4]$ misses the fact that E_0 has actually two diagonals, not three. Based on the fact, r consecutive multiplication has exactly 2^r diagonals if the first factor is E_0 and $2^{r+1} - 1$, otherwise.

Number of diagonals		E_i	E_iE_{i+1}	$E_iE_{i+1}E_{i+2}$	$E_iE_{i+1}E_{i+2}E_{i+3}$
$i=0$	split	2	5	8	11
	consecutive	2	4	8	16
$i>0$	split	3	6	9	12
	consecutive	3	7	15	31

The above table compares the number of diagonals when consecutive matrices are pre-multiplied to the number when the matrices are just split and not pre-multiplied. An optimal number of consecutive multiplication deduced from the table is three for E_0 and two for E_i with E_i with $i > 0$. For the data in $\mathbb{C}^{\frac{N}{2}}$, there are $\log \frac{N}{2}$ number of FFT factors of DFT, and an optimal number of decomposition is $\left\lceil \frac{\log \frac{N}{2}-3}{2} \right\rceil + 1 = \left\lceil \log \frac{N}{2} \right\rceil - 1$. When the available levels $l_{\mathrm{CTS}^*} + l_{\mathrm{EM}}$ for CTS^+ is bigger than the optimal number, the modulus in ciphertext is reduced via ModDown by their difference before the multiplication of FFT matrices. This allows us to decrease the remaining levels and speed up computations.

Now, we can determine the computational cost of CTS^+ in comparison to CTS^*. It's important to note that the computational complexity of a linear transformation is proportional to both the total number of diagonal vectors and the starting level. With these considerations in mind, the computational cost ratio of CTS^+ to CTS^* is given by

$$\frac{\#\text{ of diagonals in CTS}^+}{\#\text{ of diagonals in CTS}^*} \times \frac{\text{the starting level in CTS}^+}{\text{the starting level in CTS}^*}. \tag{7}$$

Given our efficient design of CTS^+, the computational cost ratio of CTS^+ to CTS^* should not result in a significant increase in computational cost.

Remark : To enhance the speed of CTS^+, our primary focus was on minimizing the number of diagonals. Additionally, we could have explored the approach of minimizing its starting level. Determining the optimal strategy between minimizing the number of diagonals and minimizing the starting level requires exhaustive trials and careful consideration of various factors, akin to the discussion presented in [11]. As depicted in Figure 7, CTS^+ is configured in parallel with the $\mathrm{CTS}^* + \mathrm{EvalMod}$. The circuits of CTS^+ and CTS^* are independent of each other and can be executed in parallel. Therefore, if the running time of CTS^+ is shorter than that of CTS^*, it does not affect the overall running time. By focusing solely on reducing the number of diagonals, CTS^+ achieves a running time of less than half of CTS^*. Hence, we can omit the complex and additional discussions mentioned above.

Example 3. In this example, we utilize Parameter I in Table 1. In this parameter set, the depth of CTS^* with Δ_{CTS^*} is 2, and the depth of EvalMod is 8, so we can utilize CTS^+ with Δ_{CTS} with 10 multiplicative depths. Since $N = 2^{16}$, there are 15 number of FFT matrices. Among 15 matrices, the first three matrices multiplied together to form one group, and then subsequent groups are formed

by sequentially multiplying two matrices. This grouping method is summarized in the below table, resulting in a total of 7 groups.

( ) : number of diagonal vectors

DFT	$E_0(2)$	$E_1(3)$	$E_2(3)$	$E_3(3)$	$E_4(3)$	$E_5(3)$	$E_6(3)$	$E_7(3)$	$E_8(3)$	$E_9(3)$	$E_{10}(3)$	$E_{11}(3)$	$E_{12}(3)$	$E_{13}(3)$	$E_{14}(3)$	total
CoeffToSlot*	$B_1(8)$			$B_2(7)$		$B_3(7)$		$B_4(7)$		$B_5(7)$		$B_6(7)$		$B_7(7)$		(50)
CoeffToSlot	B_1 (16)					B_2 (31)				B_3 (31)				B_4 (15)		(93)

Therefore, CTS$^+$ utilizes $7 = \left\lceil \frac{\log N}{2} \right\rceil - 1$ out of 10 available multiplicative depths, while the remaining 3 are consumed by ModDown before the multiplications. Let us determine the computational cost of CTS$^+$. Initially, when considering the total number of diagonal vectors, the matrices multiplied in CTS$^+$ comprise a total of 50 diagonal vectors, in contrast to the 93 in CTS*. Subsequently, in terms of starting level, CTS$^+$ starts at level 22 due to ModDown, while CTS* starts at level 25. Utilizing Equation (7), the computational cost ratio of CTS$^+$ to CTS* is given below. This implies that the computational cost of CTS$^+$ is less than half of that of CTS*.

$$\frac{\#\text{ of diagonals in CTS}^+}{\#\text{ of diagonals in CTS}^*} \times \frac{\text{the starting level in CTS}^+}{\text{the starting level in CTS}^*} = \frac{50}{93} \cdot \frac{22}{25} = 0.47\ldots$$

The table below presents a comparison of the running time of CTS$^+$ with that of CTS*. The experiments were conducted using our GitHub code with parameter set I, repeated 10 times and averaged. The ratio of 0.44 closely matches the analytical result of 0.47. This alignment between the experimental and analytical results suggests consistency and reliability in our findings.

	CTS*	CTS$^+$	CTS$^+$/CTS*
running time	247s	109s	0.44

7 Tailoring scale factors with sparsity patterns

In Example 1, observed outcomes diverged from estimations. Upon closer examination, this discrepancy stems from the recurring presence of sparsity pattern in diagonal vectors. In this section, we introduce the two terms called "sparsity pattern" and "sparsity gap" . To the best of our knowledge, the identification of this sparsity pattern is unique to our research. Recognizing these patterns has significantly improved the precision of our error analysis. Furthermore, by taking into account these sparsity patterns, we can refine the ciphertext modulus by adjusting scale factors in CTS and STC. This section delves deeper into the emergence of the sparsity pattern and its implications.

Definition 1. *Sparsity pattern is a phenomenon where zero elements appear periodically when the diagonal vectors of a matrix undergo iFFT. The period of sparsity pattern in matrix A will be denoted as $S(A)$ and it is the period of nonzero element appearance within diagonal vector.*

The sparsity pattern is characterized by the presence of consecutive zeros in the encoded diagonal vectors. Through experimentation, we have confirmed that this sparsity pattern is consistently observed across all diagonal vectors. While the underlying cause of this sparsity pattern remains elusive, our observations have led us to glean the following insights.

Conjecture 1. Let N be the RLWE dimension. Denote $E_0, \cdots, E_{\log N-2}$ as decomposed FFT matrices. Then $S\left(E_i \cdot E_{i+1} \cdots E_{i+(d-1)}\right) = S\left(E_i\right) = 2^i$ for any i and d.

The forthcoming Example 4 elucidates the observations derived from experimentation that correspond to Conjecture 1.

Example 4. Let $N = 2^5$ and scale factor $\Delta = 2^{40}$. Denote E_0, E_1, E_2 and E_3 be the decomposed FFT matrices. Each E_i has 32 diagonal vectors. We encode all 32 diagonal vectors with scale factor Δ, and note that the scale factor does not have any impact on the sparsity pattern. We observe the sparsity pattern by encoding the diagonal vectors of the matrices obtained by multiplying groups of two, three and four consecutive FFT matrices together and compute the sparsity gap. This results is summarized in Table 2. Experiment results allow us to verify the Conjecture 1, empirically.

Multiplication of FFT matrices	Sparsity Gap	Densest Matrix
E_0E_1	1	E_0
E_1E_2	2	E_1
E_2E_3	4	E_2
$E_0E_1E_2$	1	E_0
$E_1E_2E_3$	2	E_1
$E_0E_1E_2E_4$	1	E_0

Table 2. The resulting sparsity pattern resulting when multiple FFT matrices are sequentially multiplied. The results indicate that the sparsity gap of a multiplication of FFT matrices follows the sparsity gap of the densest matrix among them.

The sparsity pattern can be applied to our error analysis of linear transformations. With the incorporation of the sparsity pattern, slight adjustments are made to the error estimation formula. The new error estimation formula ensures precise prediction of experimental observations. When applied, it accurately computes the estimated values for Example 1, aligning exactly with the observed values. The forthcoming lemma demonstrates how error estimation with the consideration of the sparsity pattern emerges when a single matrix multiplication is performed.

Theorem 3. *For a matrix $A \in \mathbb{C}^{\frac{N}{2} \times \frac{N}{2}}$ and a vector $z \in \mathbb{C}^{\frac{N}{2}}$, let $Homo\left(Az\right)$ be the homomorphic evaluation of their product Az. Then its error is given as*

$$\left\| Homo\left(Az\right) - Az \right\|^2 = \frac{kN \left\| z \right\|^2}{12\Delta^2} \cdot \frac{1}{S\left(A\right)},$$

where k is the number of diagonal vectors of the matrix and Δ is the scale factor in the homomorphic evaluation.

Proof. $Az = \sum_{i=1}^{k} v_i \odot \mathsf{rot}^i\left(z\right)$, where v_i is the i^{th} diagonal vector. In the homomorphic evaluation, v_i is converted to a plaintext $\mathsf{pt}_i = \mathrm{DFT}\left(v_i \cdot \Delta\right)$ and rounded to $\widetilde{\mathsf{pt}_i} = \lfloor \mathsf{pt}_i \rceil$, generating truncation error $\tau_i\left[j\right] = \mathsf{pt}_i\left[j\right] - \widetilde{\mathsf{pt}_i}\left[j\right]$ for each $j = 0, \cdots, N-1$. Without the sparsity pattern, each $\tau_i\left[j\right]$ can be treated as a random variable with the uniform distribution $U\left[-\frac{1}{2}, \frac{1}{2}\right]$. With the pattern, $\tau_i\left[j\right] = 0$ unless j is a integer multiple of $S\left(A\right)$, since there is no truncation error of zero.

Let pt_i^z be the plaintext corresponding to $\mathsf{rot}^i\left(z\right)$, then

$$\begin{aligned} Homo\left(Az\right) - Az &= \sum_{i=1}^{k} \mathrm{DFT}\left(\tfrac{\mathsf{pt}_i}{\Delta} \cdot \tfrac{\mathsf{pt}_i^z}{\Delta}\right) - \mathrm{DFT}\left(\tfrac{\widetilde{\mathsf{pt}_i}}{\Delta} \cdot \tfrac{\mathsf{pt}_i^z}{\Delta}\right) \\ &= \sum_{i=1}^{k} \mathrm{DFT}\left(\tfrac{\tau_i}{\Delta} \cdot \tfrac{\mathsf{pt}_i^z}{\Delta}\right). \end{aligned}$$

Each component $\left(\frac{\tau_i}{\Delta} \cdot \frac{\mathsf{pt}_i^z}{\Delta}\right)\left[j\right]$ is a sum of $\frac{N}{S(A)}$ number of random variables.

$$\begin{aligned} \left(\tfrac{\tau_i}{\Delta} \cdot \tfrac{\mathsf{pt}_i^z}{\Delta}\right)\left[j\right] &= \sum_{l=0}^{N-1} \pm \tfrac{\tau_i[l]}{\Delta} \cdot \tfrac{\mathsf{pt}_i^z}{\Delta}\left[\mathrm{mod}\left(j-l,\, N\right)\right] \\ &= \sum_{S(A) \mid l} \pm \tfrac{\tau_i[l]}{\Delta} \cdot \tfrac{\mathsf{pt}_i^z}{\Delta}\left[\mathrm{mod}\left(j-l,\, N\right)\right]. \end{aligned}$$

By Lyapunov's Central limit theorem, the sum follows a normal distribution. Since N is large enough, we have

$$\begin{aligned} \left\| Homo\left(Az\right) - Az \right\|^2 &= \sum_{i=1}^{k} \tfrac{N}{2} \left\| \tfrac{\tau_i}{\Delta} \cdot \tfrac{\mathsf{pt}_i^z}{\Delta} \right\|^2 \\ &= k \cdot \tfrac{N}{2} \sum_{S(A) \mid l} \sum_{j=0}^{N-1} \tfrac{1}{12\Delta^2} \left| \tfrac{\mathsf{pt}_i^z}{\Delta}\left[\mathrm{mod}\left(j-l,\, N\right)\right] \right|^2 \\ &= k \cdot \tfrac{N}{2} \cdot \tfrac{N}{S(A)} \cdot \tfrac{1}{12\Delta^2} \cdot \left\| \tfrac{\mathsf{pt}_i^z}{\Delta} \right\|^2 \\ &= \tfrac{kN\|z\|^2}{12\Delta^2} \cdot \tfrac{1}{S(A)}. \end{aligned}$$

Here we used the L^2-isometry, $\left\| z \right\|^2 = \frac{N}{2} \cdot \left\| \frac{\mathsf{pt}_i^z}{\Delta} \right\|^2$ for each i.

When a series of matrices are multiplied, the above Theorem is recursively applied in the same way as in Theorem 2. We report the following Theorem and skip its proof, since it is a straightforward repeat of its proof.

Theorem 4. *For matrices $A_1, \cdots, A_d \in \mathbb{C}^{\frac{N}{2} \times \frac{N}{2}}$ and a vector $z \in \mathbb{C}^{\frac{N}{2}}$, let $Homo^{seq}(A_d \cdots A_1 z)$ be the homomorphic evaluation of their product $A_1, \cdots, A_d z$. Then its error is given as*

$$\|Homo^{seq}(A_d \cdots A_1 z) - A_d \cdots A_1 z\|^2 = \frac{N\|z\|^2}{12\Delta^2} \|A_1\|^2 \cdots \|A_d\|^2 \cdot \left(\frac{k_{A_1}}{\|A_1\|^2 \cdot S(A_1)} + \cdots + \frac{k_{A_d}}{\|A_d\|^2 \cdot S(A_d)} \right),$$

where k_{A_i} is the number of diagonal vectors of each matrix A_i for all i and Δ is the scale factor in the homomorphic evaluation.

The above Theorem assumed the same scale factor. When the terms $\frac{k_{A_i}}{S(A_i)}$ are not uniform, it is advantageous to take different scale factor Δ_i for each i. The error analysis is not much different with the different scale factors. For the latter case, we transform the error equation as follows.

$$\|Homo^{seq}(A_d \cdots A_1 z) - A_d \cdots A_1 z\|^2 = \frac{N\|z\|^2}{12} \|A_1\|^2 \cdots \|A_d\|^2 \cdot \left(\frac{k_{A_1}}{\|A_1\|^2 \cdot S(A_1) \cdot \Delta_1^2} + \cdots + \frac{k_{A_d}}{\|A_d\|^2 \cdot S(A_d) \cdot \Delta_d^2} \right).$$

Then, uniform scale factor Δ and adaptive scale factors $\{\Delta_i\}_{i=1}^d$ are related as below when the error sizes are equal.

$$\frac{1}{\Delta^2} \left(\frac{k_{A_1}}{\|A_1\|^2 \cdot S(A_1)} + \cdots + \frac{k_{A_d}}{\|A_d\|^2 \cdot S(A_d)} \right) = \frac{k_{A_1}}{\|A_1\|^2 \cdot S(A_1) \cdot \Delta_1^2} + \cdots + \frac{k_{A_d}}{\|A_d\|^2 \cdot S(A_d) \cdot \Delta_d^2}.$$

From the relation, we suggest selecting adaptive scale factors as follows.

$$\Delta_i = \Delta \cdot \left(d \cdot \frac{\|A_i\|^2 \cdot k(A_i)/S(A_i)}{\sum_{j=1}^{d} \|A_j\|^2 \cdot k(A_j)/S(A_j)} \right)^{1/2}, \text{ for each } i. \tag{8}$$

Example 5. In Example 1, the table comparing the observed and estimated e_{CTS^*} shows a slight discrepancy, with our estimations slightly exceeding the observed values. This example illustrates that the discrepancy can be removed by applying new error estimation formula derived from Theorem 4 and this new formula allows for much more accurate error prediction. The table below compares the observed errors of each FFT matrix multiplication in CTS with those of sparsity-applied error estimations, based on Parameter I from Table 1.

Parameter I	observation	new estimations
$\|e_{A_1}\|$	4.21×10^{-12}	4.21×10^{-12}
$\|e_{A_2A_1}\|$	4.19×10^{-13}	4.37×10^{-13}
$\|e_{A_3A_2A_1}\|$	3.96×10^{-14}	4.31×10^{-14}
$\|e_{A_4A_3A_2A_1}\|$	5.26×10^{-15}	5.99×10^{-15}

For the design of CTS with Parameter I, the sparsity gaps for A_1, A_2, A_3 and A_4 are $1, 16, 256$ and 4096 , respectively. Using these sparsity gaps in Equation (8) enables the adaptive selection of scale factors for each matrix in CTS. Similarly, the scale factors of FFT matrices multiplied within STC can also be adaptively chosen. The table compares the modulus bits used in CTS and STC before and after tailoring the scale factors for the matrices multiplied in both CTS and STC, and the maintenance of overall accuracy despite the utilization of these tailored scale factors.

Parameter I	modulus bits		$\|e_{\text{total}}\|$
	CTS	STC	
EvalRound$^+$+ Error Balancing	118.2	84.9	6.32×10^{-5}
EvalRound$^+$+ Error Balancing + Sparsity	111.6	80.4	6.34×10^{-5}

The table reveals that tailoring the scale factors for each matrix allows for approximately 7 and 4 bits reduction in CTS and STC, respectively, totaling around 11 bits saved compared to the non-tailored scenario while keeps the overall accuracy.

8 Conclusion

In this work, we introduced three methods to reduce the substantial modulus and levels consumed in conventional bootstrapping while maintaining the same precision. These methods are referred to as EvalRound$^+$, error balancing, and tailoring scale factors. They are independent of each other, allowing individual utilization. When all of them are applied in conventional settings, a significant reduction in the modulus and levels spent in CTS and STC is achieved. The saved modulus and levels are then reallocated to general-purpose multiplications (Mult). Table 3 presents the modified parameter sets I*, II*, and III* obtained by applying the proposed three methods to the conventional parameter sets I, II, and III, respectively. The modulus spent in CTS and STC is reduced by approximately 43.8%, 40.9%, and 42.4% in Parameter set I, II, and III, respectively.

By reducing the modulus and levels consumed in CTS and STC, we increase the capacity for general ciphertext multiplications (Mult) following bootstrapping. The increase in the number of Mult is proportional to the achieved reduction. For each set of parameters I, II, and III, we observe a gain of 33%, 20% and 44%, respectively, in the number of Mult compare to the conventional bootstrapping.

Note that the results are based on the assumption that the additional CTS in EvalRound$^+$ is executed independently and in parallel. Therefore, if executed

Parameter	$\log q_j$				
	q_0	Mult	STC	EvalMod	CTS
I	60	9×40	3×39	8×60	4×56
I*	60	12×40	2×40.2	8×60	2×55.8
II	55	$5 \times 60 + 5 \times 30$	2×45	8×55	4×53
II*	55	$6 \times 60 + 6 \times 30$	2×36.8	8×55	2×52.3
III	58	9×42	3×42	9×58	3×58
III*	58	13×42	2×39.8	9×58	2×46.4

Table 3. Parameter sets modified by the proposed three methods

sequentially, the efficiency is somewhat reduced. However, with the efficient implementation demonstrated in Section 6, the latency amount of the additional CTS is at most 50% of the usual CTS. Though it heavily depends on each computation environment, CTS would take about 30% charge of the total cost. With this regard, the latency of the sequential implementation would be just about 15% of the total cost. About 15% loss is bearable due to the aforementioned advantages.

From the above results, our improvements are significant for most commonly used parameter sets. Importantly, we achieved these improvements while maintaining accuracy and running time at a closely similar level to those of the conventional bootstrapping. All these experimental results are summarized in Table 4. All experiments were conducted using our self-implemented code, which will be publicly available on GitHub.

Due to the limited time available and pages allowed, we consider only CTS and STC to improve, leaving out EvalMod. Recently, there have been some noticeable enhancements of EvalMod, such as modified Remez [15] and variance minimization [16]. Our optimizations in CTS and STC can be combined with an optimization in EvalMod. We put off to future work the discussion of what other optimizations to combine with ours and how to modify the presented error analysis to these combinations.

Parameter I	Modulus consumption	General multiplications	Bootstrapping accuracy	Running time
Conventional	821	9	7.16×10^{-5}	602
EvalRound	828	10	5.00×10^{-5}	600
ours	671	12	6.34×10^{-5}	631 (parallel) 733 (serial)

Parameter II	Modulus consumption	General multiplications	Bootstrapping accuracy	Running time
Conventional	742	10	5.36×10^{-2}	692
EvalRound	752	10	4.81×10^{-2}	700
ours	617	12	7.59×10^{-2}	647 (parallel) 750 (serial)

Parameter III	Modulus consumption	General multiplications	Bootstrapping accuracy	Running time
Conventional	822	9	1.41×10^{-5}	1075
EvalRound	813	10	1.32×10^{-5}	1068
ours	693	13	1.40×10^{-5}	1070 (parallel) 1167 (serial)

Table 4. When comparing our proposal, which is obtained by applying the proposed three methods, to conventional and EvalRound bootstrappings, ours exhibits the least bit consumption of modulus and the highest number of general multiplications per bootstrapping. However, the accuracy and runtime (in seconds) of bootstrapping are similar among the three methods. Accuracy refers to $\|e_{total}\|$ in Section 5.

References

1. Bae, Y., Cheon, J.H., Cho, W., Kim, J., Kim, T.: Meta-bts: Bootstrapping precision beyond the limit. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. pp. 223–234. CCS '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3548606.3560696, https://doi.org/10.1145/3548606.3560696
2. Bae, Y., Cheon, J.H., Kim, J., Park, J.H., Stehlé, D.: Hermes: Efficient ring packing using mlwe ciphertexts and application to transciphering. In: Annual International Cryptology Conference. pp. 37–69. Springer (2023)
3. Billingsley, P.: Probability and Measure. Wiley Series in Probability and Statistics, Wiley (2012), https://books.google.co.kr/books?id=a3gavZbxyJcC
4. Bossuat, J.P., Mouchet, C., Troncoso-Pastoriza, J., Hubaux, J.P.: Efficient bootstrapping for approximate homomorphic encryption with non-sparse keys. In: Advances in Cryptology–EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part I. pp. 587–617. Springer (2021)
5. Chen, H., Chillotti, I., Song, Y.: Improved bootstrapping for approximate homomorphic encryption. In: IACR Cryptology ePrint Archive (2019), https://api.semanticscholar.org/CorpusID:53240997
6. Cheon, J.H., Han, K., Kim, A., Kim, M., Song, Y.: Bootstrapping for approximate homomorphic encryption. In: Advances in Cryptology–EUROCRYPT 2018: 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29-May 3, 2018 Proceedings, Part I 37. pp. 360–384. Springer (2018)
7. Cheon, J.H., Han, K., Kim, A., Kim, M., Song, Y.: A full rns variant of approximate homomorphic encryption. Selected areas in cryptography : annual international workshop, SAC proceedings. SAC **11349**, 347–368 (2018), https://api.semanticscholar.org/CorpusID:52977564
8. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23. pp. 409–437. Springer (2017)
9. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing. pp. 169–178. STOC '09, Association for Computing Machinery, New York, NY, USA (2009). https://doi.org/10.1145/1536414.1536440, https://doi.org/10.1145/1536414.1536440
10. Halevi, S., Shoup, V.: Bootstrapping for helib. J. Cryptol. **34**(1) (jan 2021). https://doi.org/10.1007/s00145-020-09368-7, https://doi.org/10.1007/s00145-020-09368-7
11. Han, K., Hhan, M., Cheon, J.H.: Improved homomorphic discrete fourier transforms and fhe bootstrapping. IEEE Access **7**, 57361–57370 (2019). https://doi.org/10.1109/ACCESS.2019.2913850
12. Han, K., Ki, D.: Better bootstrapping for approximate homomorphic encryption. In: Jarecki, S. (ed.) Topics in Cryptology – CT-RSA 2020. pp. 364–390. Springer International Publishing, Cham (2020)

13. Jutla, C.S., Manohar, N.: Sine series approximation of the mod function for bootstrapping of approximate he. In: Dunkelman, O., Dziembowski, S. (eds.) Advances in Cryptology – EUROCRYPT 2022. pp. 491–520. Springer International Publishing, Cham (2022)
14. Kim, S., Park, M., Kim, J., Kim, T., Min, C.: Evalround algorithm in ckks bootstrapping. In: Advances in Cryptology–ASIACRYPT 2022: 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5–9, 2022, Proceedings, Part II. pp. 161–187. Springer (2023)
15. Lee, J.W., Lee, E., Lee, Y., Kim, Y.S., No, J.S.: High-precision bootstrapping of rns-ckks homomorphic encryption using optimal minimax polynomial approximation and inverse sine function. In: Canteaut, A., Standaert, F.X. (eds.) Advances in Cryptology – EUROCRYPT 2021. pp. 618–647. Springer International Publishing, Cham (2021)
16. Lee, Y., Lee, J.W., Kim, Y.S., Kim, Y., No, J.S., Kang, H.: High-precision bootstrapping for approximate homomorphic encryption by error variance minimization. In: Dunkelman, O., Dziembowski, S. (eds.) Advances in Cryptology – EUROCRYPT 2022. pp. 551–580. Springer International Publishing, Cham (2022)
17. Remez, E.Y.: Sur la determination des polynomes d'approximation de degre donnee. Comm. Soc. Math. Kharkov **10**(196), 41–63 (1934)
18. Winkelbauer, A.: Moments and absolute moments of the normal distribution. arXiv preprint arXiv:1209.4340 (2012)

Appendix

Lemma 1. *Let $\tau \in \mathbb{R}^N$ be an i.i.d. random vector with mean 0 and variance σ^2 and $z \in \mathbb{C}^{N/2}$ be a given vector. Then we have*

$$\mathbb{E}\left[\sum_{i=1}^{N/2} \left[DFT(\tau) \odot z\right]_i^2\right] = N\sigma^2 \left\|z\right\|^2.$$

Furthermore, each $\left[DFT(\tau) \odot z\right]_i$ follows the normal distribution $\mathcal{N}\left(0, N\sigma^2 \left|z_i\right|^2\right)$.

Proof. Let $\mathsf{pt}^z = \mathrm{iDFT}(z) \in \mathbb{R}[x]/x^N + 1$. Then,

$$\mathrm{DFT}(\tau) \odot z = \mathrm{DFT}(\tau \cdot \mathsf{pt}^z)$$

Each component $(\tau \cdot \mathsf{pt}^z)[j]$ is a weighted sum of i.i.d. random variables.

$$(\tau \cdot \mathsf{pt}^z)[j] = \sum_{l=0}^{N-1} \pm\tau[l] \cdot \mathsf{pt}^z[\mathrm{mod}(j-l,\, N)]$$

By Lyapunov's Central limit theorem [3], the sum follows a normal distribution with mean zero. The variance of the distribution is given as

$$\begin{aligned}
\mathbb{E}\left[\left\|(\tau \cdot \mathsf{pt}^z)[j]\right\|^2\right] &= \sigma^2 \sum_{l=0}^{N-1} \left|\mathsf{pt}^z[l]\right|^2 = \sigma^2 \left\|\mathsf{pt}^z\right\|^2 \\
\mathbb{E}\left[\left\|(\tau \cdot \mathsf{pt}^z)\right\|^2\right] &= \sigma^2 N \left\|\mathsf{pt}^z\right\|^2
\end{aligned}$$

Using the L^2-isometry, we obtain

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=1}^{N/2} \left[\mathrm{DFT}(\tau) \odot z\right]_i^2\right] &= \frac{N}{2}\mathbb{E}\left[\left\|\tau \cdot \mathsf{pt}^z\right\|^2\right] \\
&= \frac{N}{2}\sigma^2 N \left\|\mathsf{pt}^z\right\|^2 \\
&= \sigma^2 N \left\|z\right\|^2.
\end{aligned}$$

Proof of Theorem 1 :

Proof. Let $v_1, \cdots, v_k$ be the diagonal vectors of A so that $Az = \sum_{j=1}^{k} v_j \odot z_j$, where z_j is a rotation of z. Let $\mathsf{pt}_j = \Delta \cdot \mathrm{iDFT}(v_j)$, then the encoding vector of $v_j$equals $\lfloor \mathsf{pt}_j \rceil$ and $\tau_j = \lfloor \mathsf{pt}_j \rceil - \mathsf{pt}_j$ follows the uniform distribution $U\left[-\frac{1}{2}, \frac{1}{2}\right]$, for each j. Then we have

$$
\begin{aligned}
e &= \frac{1}{\Delta}\sum_{j=1}^{k}\mathrm{DFT}\left(\lfloor \mathsf{pt}_j \rceil\right)\odot z_j - \sum_{j=1}^{k} v_j \odot z_j \\
&= \frac{1}{\Delta}\sum_{j=1}^{k}\mathrm{DFT}\left(\lfloor \mathsf{pt}_j \rceil\right)\odot z_j - \frac{1}{\Delta}\sum_{j=1}^{k}\mathrm{DFT}\left(\mathsf{pt}_j\right)\odot z_j \\
&= \frac{1}{\Delta}\sum_{j=1}^{k}\mathrm{DFT}\left(\tau_j\right)\odot z_j.
\end{aligned}
$$

Let w_j denote $\mathrm{DFT}\left(\tau_j\right)\odot z_j$, then $\mathbb{E}\left[\|w_j\|^2\right] = \frac{N}{12}\|z\|^2$ for each j. Since the rounding errors $\tau_1, \cdots, \tau_k$ are independent to each other, so are $w_1, \cdots, w_k$. Using the independence, we can use the additivity of variance to get $\mathbb{E}\left[\|e\|^2\right] = \frac{1}{\Delta^2}\sum_{j=1}^{k}\mathbb{E}\left[\|w_j\|^2\right] = \frac{kN}{12\Delta^2}\|z\|^2$. By the above lemma, the i^{th} coordinate of w_j, denoted by $w_{j,i}$, follows the normal distribution $\mathcal{N}\left(0, \frac{N}{12}|z_{j,i}|^2\right)$ and their sum e_i follows the normal distribution $\mathcal{N}\left(0, \frac{N}{12\Delta^2}\sum_{j=1}^{k}|z_{j,i}|^2\right)$. The variance is at most $\frac{N}{12\Delta^2}k\|z\|_\infty^2$.

Proof of Theorem 2 :

Proof. Let us enumerate the errors of single matrix multiplications.

$$
\begin{aligned}
e_1 &= Homo\left(A_1 z\right) - A_1 z \\
&\vdots \\
e_{d-1} &= Homo\left(A_{d-1}Homo^{seq}\left(A_{d-2}\cdots A_1 z\right)\right) - A_{d-1}Homo^{seq}\left(A_{d-2}\cdots A_1 z\right) \\
e_d &= Homo\left(A_d Homo^{seq}\left(A_{d-1}\cdots A_1 z\right)\right) - A_d Homo^{seq}\left(A_{d-1}\cdots A_1 z\right)
\end{aligned}
$$

The magnitude of each error is estimated by the previous theorem.

$$
\begin{aligned}
\mathbb{E}\left[\|e_1\|^2\right] &= \frac{k_1 N}{12\Delta^2}\|z\|^2 \\
&\vdots \\
\mathbb{E}\left[\|e_{d-1}\|^2\right] &= \frac{k_{d-1}N}{12\Delta^2}\|Homo^{seq}\left(A_{d-2}\cdots A_1 z\right)\|^2 \vdots \\
\mathbb{E}\left[\|e_d\|^2\right] &= \frac{k_d N}{12\Delta^2}\|Homo^{seq}\left(A_{d-1}\cdots A_1 z\right)\|^2
\end{aligned}
$$

Note that the first term of e and e_d cancels out and $e - e_d$ has the common factor A_d as follows.

$$
\begin{aligned}
e - e_d =& A_d Homo^{seq}\left(A_{d-1}\cdots A_1 z\right) - A_d A_{d-1}\cdots A_1 z \\
=& A_d\left[Homo^{seq}\left(A_{d-1}\cdots A_1 z\right) - A_{d-1}\cdots A_1 z\right]
\end{aligned}
$$

Likewisely, the first term of $e-e_d$ and $A_d e_{d-1}$ cancels out and $e-e_d-A_d e_{d-1}$ has the common factor $A_d A_{d-1}$ as follows.

$$
\begin{aligned}
e - e_d - A_d e_{d-1} =& A_d A_{d-1} Homo^{seq}\left(A_{d-2}\cdots A_1 z\right) - A_d A_{d-1} A_{d-2}\cdots A_1 z \\
=& A_d A_{d-1}\left[Homo^{seq}\left(A_{d-2}\cdots A_1 z\right) - A_{d-2}\cdots A_1 z\right]
\end{aligned}
$$

Continuing this approach leads to the following decomposition of e.

$$
e = e_d + A_d e_{d-1} + A_d A_{d-1} e_{d-2} + \cdots + A_d A_{d-1}\cdots A_2 e_1
$$

Since the truncation errors are independent to each other, the decomposition is a sum of independent random variables. Using the independence and the conformality, we get

$$
\begin{aligned}
\mathbb{E}\left[\|e\|^2\right] =& \mathbb{E}\left[\|e_d\|^2\right] + \mathbb{E}\left[\|A_d e_{d-1}\|^2\right] + \cdots + \mathbb{E}\left[\|A_d A_{d-1}\cdots A_2 e_1\|^2\right] \\
=& \mathbb{E}\left[\|e_d\|^2\right] + \|A_d\|^2 \mathbb{E}\left[\|e_{d-1}\|^2\right] + \cdots + \|A_d\|^2 \cdots \|A_2\|^2 \mathbb{E}\left[\|e_1\|^2\right] \\
=& \frac{k_d N}{12\Delta^2}\|A_{d-1}\cdots A_1 z\|^2 \\
& + \|A_d\|^2 \frac{k_{d-1} N}{12\Delta^2}\|A_{d-2}\cdots A_1 z\|^2 \\
& + \cdots \\
& + \|A_d\|^2 \cdots \|A_2\|^2 \frac{k_1 N}{12\Delta^2}\|z\|^2 \\
=& \frac{N\|z\|^2}{12\Delta^2}\|A_d\|^2 \cdots \|A_1\|^2 \left(\frac{k_d}{\|A_d\|^2} + \cdots + \frac{k_1}{\|A_1\|^2}\right).
\end{aligned}
$$