Minimax Approximation of Sign Function by Composite Polynomial for Homomorphic Comparison

Eunsang Lee[1], Joon-Woo Lee[1], Jong-Seon No[1], and Young-Sik Kim[2]

[1] Seoul National University, Republic of Korea
eslee3209@ccl.snu.ac.kr, joonwoo3511@ccl.snu.ac.kr, jsno@snu.ac.kr
[2] Chosun University, Republic of Korea
iamyskim@Chosun.ac.kr

Abstract. The comparison function of the two numbers is one of the most commonly used operations in many applications including deep learning and data processing systems. Several studies have been conducted to efficiently evaluate the comparison function in homomorphic encryption schemes which only allow addition and multiplication for the ciphertext. Recently, new comparison methods that approximate sign function using composite polynomial in the homomorphic encryption, called homomorphic comparison operation, were proposed and it was proved that the methods have optimal asymptotic complexity. In this paper, we propose new optimal algorithms that approximate the sign function in the homomorphic encryption by using composite polynomials of the minimax approximate polynomials, which are constructed by the modified Remez algorithm. It is proved that the number of required non-scalar multiplications and depth consumption for the proposed algorithms are less than those for any methods that use a composite polynomial of component polynomials with odd degree terms approximating the sign function, respectively. In addition, an optimal polynomial-time algorithm for the proposed homomorphic comparison operation is proposed by using dynamic programming. As a result of numerical analysis, for the case that we want to minimize the number of non-scalar multiplications, the proposed algorithm reduces the required number of non-scalar multiplications and depth consumption by about 33% and 35%, respectively, compared to those for the previous work. In addition, for the case that we want to minimize the depth consumption, the proposed algorithm reduces the required number of non-scalar multiplications and depth consumption by about 10% and 47%, respectively, compared to those for the previous work.

Keywords: Cheon-Kim-Kim-Song (CKKS) scheme · fully homomorphic encryption (FHE) · homomorphic comparison operation · minimax approximate polynomial · Remez algorithm · sign function.

1 Introduction

Homomorphic encryption (HE) is a cryptographic algorithm that allows algebraic operations over the encrypted data. Until Gentry's seminal work [2] in

2009, HE schemes were able to perform only a few specific operations for the encrypted data. Fully homomorphic encryption (FHE) is a cryptographic algorithm that allows all algebraic operations on the encrypted data without restriction and a FHE scheme was first developed in [2]. Due to the feature, FHE has attracted significant attention in various applications and the standardization process for FHE is in progress.

FHE schemes can be classified as bit-wise FHE and word-wise FHE. Word-wise FHE such as Brakerski/Fan-Vercauteren (BFV) [4] and Cheon-Kim-Kim-Song (CKKS) [6] provides the addition and multiplication of an encrypted array over $\mathbb{C}$ or $\mathbb{Z}_p$ for a positive integer $p > 2$. All other operations in word-wise FHE should be performed using these two basic operations. On the other hand, the basic operations of bit-wise FHEs such as the fast fully homomorphic encryption over the torus (TFHE) [5] are logic gates such as NAND gates. Recently, word-wise FHE has been widely used in many applications such as deep learning [21, 22].

The comparison function is denoted as $\mathrm{comp}(a, b)$, which outputs 1 if $a > b$, $1/2$ if $a = b$, and 0 if $a < b$. The comparison function is one of the most commonly used operations along with addition and multiplication in many applications including machine learning algorithms [19, 20]. However, when we encrypt inputs word-wise, it is known to be difficult to perform the comparison operation for the ciphertexts in FHEs, called a homomorphic comparison operation, since the comparison operation is a non-polynomial operation. Thus, it is indispensable to find an efficient method to implement the homomorphic comparison operation.

In this paper, a new efficient method to perform the homomorphic comparison operation in word-wise FHEs is proposed. Since comparison operation is a non-polynomial operation, it is necessary to find and evaluate a polynomial that approximates $\mathrm{comp}(a, b)$. Comparison operations can be implemented by sign function, that is, $\mathrm{comp}(a, b) = \frac{1}{2}(\mathrm{sgn}(a - b) + 1)$. Thus, in order to perform a homomorphic comparison operation, it is enough to find a polynomial that well approximates $\mathrm{sgn}(x)$.

It is desirable to find the approximate polynomial that requires the minimum computational complexity and depth consumption while satisfying a given approximation error bound. Addition, scalar multiplication, and non-scalar multiplication affect the computational complexity. However, non-scalar multiplication requires the largest computational complexity by far. Although the efficiency of FHE has been improved a lot, non-scalar multiplication still requires a considerable amount of computational complexity. Thus, a polynomial approximation of $\mathrm{sgn}(x)$, which minimizes the number of non-scalar multiplications and depth consumption, is proposed in this paper.

1.1 Previous Works

Some research has been done on how to find polynomials that approximate the sign function $\mathrm{sgn}(x)$ or $\mathrm{comp}(a, b)$ in FHE. An analytic method to approximate the sign function using the Fourier series was proposed in [17]. In [18], the sign function was approximated using the approximate equation $\tanh(kx) =$

$\frac{e^{kx}-e^{-kx}}{e^{kx}+e^{-kx}} \simeq \mathrm{sgn}(x)$ for large $k > 0$. Recently, an iterative algorithm was proposed that performs homomorphic comparison operation using the equation $\lim_{k\to\infty} \frac{a^k}{a^k+b^k} = \mathrm{comp}(a, b)$ in [7], where the inverse operation can be performed using Goldschmidt's division algorithm [15]. However, the use of inverse operation causes some inefficiency in computational complexity. More recently, the homomorphic comparison operation is approximated using composite polynomial with less non-scalar multiplications and depth consumption than the previous methods in [8]. It was also shown that the homomorphic comparison operation by using composite polynomial has optimal asymptotic computational complexity. However, the performance of the homomorphic comparison operation using composite polynomials in [8] can be further improved since the composite polynomials used in [8] do not guarantee optimality for the approximation of the sign function by polynomials. Although there have been some improvements, the homomorphic comparison operation still requires a lot of time, and thus more research is needed to improve the performance of the homomorphic comparison operation for practical use.

1.2 Our Contributions

In this paper, we propose that if composite polynomials of component minimax approximate polynomials obtained by the modified Remez algorithm [11] are used, the efficiency of the homomorphic comparison operation can be further improved, where we have three contributions as follows.

First, we propose a method of approximating the sign function with composite polynomials of component minimax approximate polynomials. Our main idea is to find the composite polynomial which minimizes the non-scalar multiplications and depth consumption among all of the composite polynomials of component minimax approximate polynomials.

Second, since the sign function is an odd function, it is natural to use composite polynomials consisting of component polynomials with only odd degree terms. All the component polynomials used in [8] are also polynomials with odd degree terms. It is proved that the composite polynomials of component polynomials with odd degree terms found by the proposed method is the best among all of the composite polynomials of component polynomials with odd degree terms. That is, the composite polynomial obtained by the proposed method requires less number of non-scalar multiplications and depth consumption than any other composite polynomials of component polynomials with odd degree terms.

Third, even though the optimal composite polynomials of component minimax approximate polynomials can be found by the brute-force search from the candidate composite polynomials of component minimax approximate polynomials, the brute-force search requires an exponential time with respect to α, which corresponds to bit precision. Thus, polynomial-time algorithms using dynamic programming which find the optimal composite polynomials in polynomial time are proposed. By using the dynamic programming, the number of required non-scalar multiplications and depth consumption for evaluation of the proposed

composite polynomials for the homomorphic comparison operation are obtained and compared to those for the previous method [8]. It can be seen that for the case that we want to minimize the number of non-scalar multiplications, the proposed algorithm reduces the required number of non-scalar multiplications and depth consumption by about 33% and 35%, respectively, compared to those for the previous algorithm. In addition, for the case that we want to minimize the depth consumption, the proposed algorithm reduces the required number of non-scalar multiplications and depth consumption by about 10% and 47%, respectively, compared to those for the previous work.

1.3 Outline

The outline of the paper is given as follows. Section 2 presents some preliminaries for the concept of FHE, comparison operation in FHE, approximation theory, and the algorithms for minimax approximation. In Section 3, a new method to approximate the sign function using composite polynomial of minimax approximate polynomials is proposed and it is proved that the proposed method of approximating the sign function using composite polynomial of minimax approximate polynomials is optimal. In addition, a polynomial-time algorithm to obtain the best composite polynomial for the homomorphic comparison operation is proposed by using dynamic programming. In Section 4, the numerical results of the proposed homomorphic comparison operation are given for both when the number of non-scalar multiplications is minimized and when the depth consumption is minimized. Finally, the concluding remarks are given in Section 5.

2 Preliminaries

2.1 Fully Homomorphic Encryption

In the IoT era, a lot of devices communicate over the Internet. A third party will inevitably be asked to process the data because many devices cannot process data on their own. However, if the data to be processed is confidential and the third party is unreliable, the data should be sent encrypted, and the third party should perform operations on the encrypted data. HE allows operations over the encrypted data without decryption for this case.

Until Gentry's seminal work [2] in 2009, HE schemes were able to perform only a few specific operations on the encrypted data. FHE is a cryptosystem that can perform infinite number of algebraic operations on the encrypted data with bootstrapping. A FHE scheme was first developed in [2] and many FHE schemes have since been proposed to improve efficiency [4–6]. From now on, we will consider only the FHE rather than the HE.

FHE schemes are classified as bit-wise FHE and word-wise FHE. The basic operations of bit-wise FHE are logic gates, and the basic operations of word-wise FHE are algebraic operations such as addition and multiplication. In this paper,

we focus only on word-wise FHE and thus the FHE is used instead of word-wise FHE. The definition of FHE is given as follows.

Definition 1. *A FHE scheme E is a set of five polynomial-time algorithms that satisfy the followings:*

- $\mathsf{KeyGen}(\lambda) \rightarrow (\mathrm{pk}, \mathrm{sk})$*;* KeyGen *takes security parameter* λ *as an input and outputs public key* pk *and secret key* sk*.*
- $\mathsf{Enc}(\mu, \mathrm{pk}) \rightarrow \mathrm{ct}$*;* Enc *takes a public key* pk *and a message* μ *as inputs, and outputs a ciphertext* ct *of* μ*.*
- $\mathsf{Dec}(\mathrm{ct}, \mathrm{sk}) \rightarrow \mu'$ *or* $\perp$*;* Dec *takes a ciphertext* ct *and a secret key* sk *as inputs, and outputs a message* μ'*. If the decryption procedure fails,* Dec *outputs a special symbol* $\perp$*.*
- $\mathsf{Add}(\mathrm{ct}_1, \mathrm{ct}_2, \mathrm{evk})$*;* Add *takes ciphertexts* ct_1 *and* ct_2 *of* μ_1 *and* μ_2*, respectively, and an evaluation key* evk *as inputs, and outputs a ciphertext* $\mathrm{ct}_{\mathrm{add}}$ *of* $\mu_1 + \mu_2$*.*
- $\mathsf{Mult}(\mathrm{ct}_1, \mathrm{ct}_2, \mathrm{evk})$*;* Mult *takes ciphertexts* ct_1 *and* ct_2 *of* μ_1 *and* μ_2*, respectively, and an evaluation key* evk *as inputs, and outputs a ciphertext* $\mathrm{ct}_{\mathrm{mult}}$ *of* $\mu_1 \cdot \mu_2$*.*

In CKKS scheme, there are two kinds of multiplications: scalar multiplication and non-scalar multiplication. Non-scalar multiplications require much more computational complexity than scalar multiplications. Thus, in this paper, when the homomorphic comparison operation is considered, we focus on reducing the number of non-scalar multiplications rather than scalar multiplications, together with depth consumption.

2.2 Comparison Operation in Fully Homomorphic Encryption

FHEs support addition and multiplication operations on the encrypted data, but do not support any non-arithmetic operations such as comparison operation. Thus, the approximation of comparison operation should be performed by using addition and multiplication operations in FHE. The comparison function and sign function are denoted as

$$\mathrm{comp}(a,b) = \begin{cases} 1 & \text{if } a > b \\ 1/2 & \text{if } a = b \\ 0 & \text{if } a < b \end{cases}, \quad \mathrm{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}.$$

Our goal is to perform approximation for $\mathrm{comp}(a,b)$, which is implemented only with additions and multiplications. Note that $\mathrm{comp}(a,b)$ and $\mathrm{sgn}(x)$ functions have the following relationships as

$$\mathrm{sgn}(x) = 2\mathrm{comp}(x,0) - 1, \ \mathrm{comp}(a,b) = \frac{\mathrm{sgn}(a-b)+1}{2}.$$

Thus, the approximation of $\mathrm{comp}(a,b)$ is equivalent to that of $\mathrm{sgn}(x)$. Therefore, we only focus on the polynomial approximation for $\mathrm{sgn}(x)$.

Even though the efficiency of FHEs has been improved a lot since the first FHE was developed in 2009, it is known that the non-scalar multiplication operation still takes a lot of computational complexity. In addition, since bootstrapping requires a lot of computational complexity, minimizing the depth consumption for the homomorphic comparison operation is also important, which reduces the number of bootstrappings. Thus, it is necessary to approximate $\mathrm{sgn}(x)$ by polynomials while minimizing the number of non-scalar multiplications and depth consumption.

Definition 2 ([8]). *For $\alpha > 0$ and $0 < \epsilon < 1$, a polynomial p is said to be (α, ϵ)-close to $\mathrm{sgn}(x)$ over $[-1, 1]$ if p satisfies the following:*

$$||p(x) - \mathrm{sgn}(x)||_{\infty,[-1,-\epsilon]\cup[\epsilon,1]} \leq 2^{-\alpha},$$

where $||\cdot||_{\infty,D}$ denotes the infinity norm over the domain D.

$\mathrm{sgn}(x)$ is discontinuous at $x = 0$, and thus it is impossible to exactly approximate $\mathrm{sgn}(x)$ near $x = 0$. Definition 2 means that the approximation error is guaranteed below $2^{-\alpha}$ only for $\epsilon \leq |x| \leq 1$. Figure 1 shows an example of a function satisfying (α, ϵ)-close.

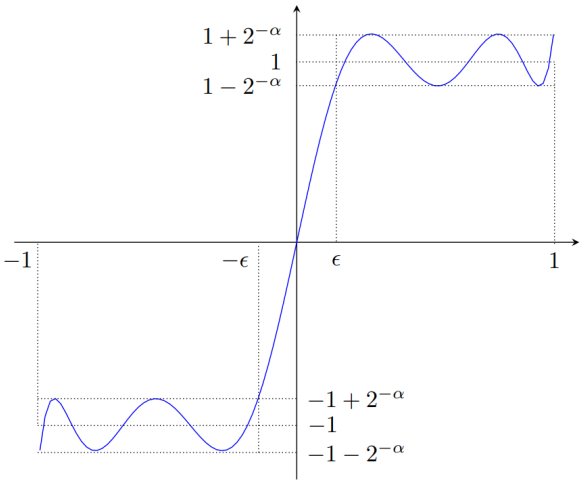

Fig. 1. An example of an approximate polynomial satisfying (α, ϵ)-close for sign function.

2.3 Approximation Theory

In this section, some concepts for approximation theory are introduced.

Definition 3. *Let D be a closed subset of $[a, b]$. Let f be a continuous function on D. A polynomial p is said to be the minimax approximate polynomial of degree at most n on D for f if p minimizes $max_D||p(x) - f(x)||_\infty$ among polynomials of degree at most n.*

It is known that for any continuous function f on D, the minimax approximate polynomial of degree at most n on D for f uniquely exists [12]. We put $f(x) = \mathrm{sgn}(x)$ since the goal in this paper is to approximate $\mathrm{sgn}(x)$. We also only deal with cases where D is the union of two symmetric closed intervals, $[-b, -a] \cup [a, b]$.

Definition 4 (Haar's Condition and Generalized Polynomial [12]). *A set of functions $\{g_1, g_2, \cdots, g_n\}$ satisfies the Haar's condition if each g_i is continuous function and if the determinant*

$$D[x_1, \cdots, x_n] = \begin{vmatrix} g_1(x_1) & \cdots & g_n(x_1) \\ \vdots & \ddots & \vdots \\ g_1(x_n) & \cdots & g_n(x_n) \end{vmatrix}$$

is not zero for any n distinct points $x_1, \cdots, x_n$. A linear combination of $\{g_1, \cdots, g_n\}$ is referred to as a generalized polynomial.

The following theorem and lemmas are needed for some proofs in Section 3.

Theorem 1 (Chebyshev Alternation Theorem [12]). *Let D be a closed subset of $[a, b]$. Let $\{g_1, g_2, \cdots, g_n\}$ be a set of continuous functions on $[a, b]$ which satisfies the Haar's condition. A polynomial $p = \sum_i c_i g_i$ is the minimax approximate polynomial on D to any given continuous function f on D if and only if there are $n+1$ elements $x_0 < \cdots < x_n$ in D such that $r(x_i) = -r(x_{i-1}) = \pm||r||_\infty$, $1 \leq i \leq n$ for the error function $r = f - p$.*

Remark 1. Let D be $[-b, -a] \cup [a, b]$. Since $r(x_i) = \pm||r||_\infty$ for $0 \leq i \leq n$, $r(x)$ should have extreme points at x_i for $0 \leq i \leq n$. Thus, it holds that $p'(x_i) = 0$ and $x_i \in (-b, -a) \cup (a, b)$, or $x_i \in \{-b, -a, a, b\}$.

Lemma 1 (Generalized de La Vallee Poussin Theorem [11]). *Let $\{g_1, g_2, \cdots, g_n\}$ be a set of continuous functions on $[a, b]$ that satisfies the Haar's condition. Let D be a closed subset of $[a, b]$ and let $f(x)$ be a continuous function on D. Let $x_i, 0 \leq i \leq n$ be $n + 1$ consecutive points on D. Let $p(x)$ be a generalized polynomial such that $p - f$ has alternately positive and negative values at $x_i, 0 \leq i \leq n$. Let $p^*(x)$ be a minimax approximate polynomial on D for f and let $e(f)$ be the minimax approximaton error of $p^*(x)$. Then, it holds that*

$$e(f) \geq \min_i |p(x_i) - f(x_i)|.$$

Lemma 2 ([16]). *If $f(x)$ is an odd function, the minimax approximate polynomial of degree at most n to $f(x)$ is also odd function.*

2.4 Algorithms for Minimax Approximation

Remez algorithm [10] obtains the minimax approximate polynomials of a continuous function on one interval. It was proved that the Remez can always find the exact minimax approximate polynomials.

Algorithm 1: Remez algorithm [11]

Input: Polynomial basis $\{g_1, \cdots, g_n\}$, a domain $[a, b]$, an approximation parameter δ, and a continuous function f on $[a, b]$
Output: The minimax approximate polynomial p for f

1 Choose $x_1, \cdots, x_{n+1} \in [a, b]$, where $x_1 < \cdots < x_{n+1}$;
2 Find the polynomial $p(x)$ in terms of $\{g_1, \cdots, g_n\}$ such that $p(x_i) - f(x_i) = (-1)^i E, 1 \leq i \leq n+1$ for some E;
3 Divide the domain $[a, b]$ into $n+1$ sections $[z_{i-1}, z_i], i = 1, \cdots, n+1$. $z_1, \cdots, z_n$ are zeros of $p(x) - f(x)$, where $x_i < z_i < x_{i+1}$, and $z_0 = a, z_{n+1} = b$;
4 Find the maximum or minimum point for each section when $p(x_i) - f(x_i)$ has positive or negative value, respectively. These points $y_1, \cdots, y_{n+1}$ are called extreme points;
5 $\epsilon_{max} \leftarrow \max_{1 \leq i \leq n+1} |p(y_i) - f(y_i)|$;
6 $\epsilon_{min} \leftarrow \min_{1 \leq i \leq n+1} |p(y_i) - f(y_i)|$;
7 **if** $(\epsilon_{max} - \epsilon_{min})/\epsilon_{min} < \delta$ **then**
8 Return $p(x)$;
9 **else**
10 Replace x_i's with y_i's. Go to line 2;
11 **end**

Recently, Lee et al. [11] proposed a modified Remez algorithm which finds the minimax approximate polynomial on multiple intervals and proved that the algorithm can always find the minimax approximate polynomial for any piecewise continuous function. This modified Remez algorithm is used in this paper to find the minimax approximate polynomial for the sign function.

Let $\mu(x)$ be a function defined as

$$\mu(x) = \begin{cases} 1 & p(x) - f(x) \text{ is a local maximum value at } x \text{ on } D \\ -1 & p(x) - f(x) \text{ is a local minimum value at } x \text{ on } D \\ 0 & \text{otherwise.} \end{cases}$$

There are three criteria for choosing $n+1$ extreme points in Algorithm 2 as follows:

(i) Local extreme value condition; $\min_i \mu(y_i)(p(y_i) - f(y_i)) \geq E$.
(ii) Alternating condition; $\mu(y_i) \cdot \mu(y_{i+1}) = -1$ for $i = 1, \cdots, n$.

Algorithm 2: Modified Remez algorithm [11]

Input: A polynomial basis $\{g_1, \cdots, g_n\}$, an approximation parameter δ, an input domain $D = \bigcup_{i=1}^{l}[a_i, b_i] \subset \mathbb{R}$, and a continuous function f on D

Output: The minimax approximate polynomial p for f

1 Choose $x_1, \cdots, x_{n+1} \in D$, where $x_1 < \cdots < x_{n+1}$;
2 Find the polynomial $p(x)$ in terms of $\{g_1, \cdots, g_n\}$ such that $p(x_i) - f(x_i) = (-1)^i E, 1 \le i \le n+1$ for some E;
3 Collect all the extreme and boundary points such that $\mu(x)(p(x) - f(x)) \ge |E|$ and put them in a set B;
4 Find $n+1$ extreme points $y_1 < y_2 < \cdots < y_{n+1}$ in B which satisfy alternating condition and maximum absolute sum condition;
5 $\epsilon_{max} \leftarrow \max_{1 \le i \le n+1} |p(y_i) - f(y_i)|$;
6 $\epsilon_{min} \leftarrow \min_{1 \le i \le n+1} |p(y_i) - f(y_i)|$;
7 **if** $(\epsilon_{max} - \epsilon_{min})/\epsilon_{min} < \delta$ **then**
8 Return $p(x)$;
9 **else**
10 Replace x_i's with y_i's. Go to line 2;
11 **end**

(iii) Maximum absolute sum condition; $\sum_{i=1}^{n+1} |p(y_i) - f(y_i)|$ is maximum for all candidate set of extreme points satisfying the local extreme value condition and the alternating condition.

The modified Remez algorithm operates with n basis functions $\{g_1, g_2, \cdots, g_n\}$. Suppose that the minimax approximate polynomial $p(x)$ is represented with the basis functions as $p(x) = \sum_{i=1}^{n} c_i g_i(x)$. The modified Remez algorithm finds the coefficients c_i's of $p(x)$. The simplest basis functions are a power basis, $\{1, x, x^2, \cdots, x^{n-1}\}$. However, when approximating the sign function using this basis, the magnitudes of the coefficients c_i's are unstable such as too small values or too large values, which makes a lot of numerical errors. Therefore, the Chebyshev polynomials are usually used as the basis functions. The Chebyshev polynomials T_i's on $[-1, 1]$ are defined by the following recursion;

$$
\begin{aligned}
&T_0(t) = 1 \\
&T_1(t) = t \\
&T_i(t) = 2tT_{i-1}(t) - T_{i-2}(t) \text{ for } i \ge 2.
\end{aligned}
$$

If the sign function is approximated on a domain $[-b, b]$ for some $b > 1$, then $\tilde{T}_i(t) = T_i(t/b)$ should be used instead of T_i for all i.

3 Approximation of Sign Function by Using Optimal Composition of Minimax Approximate Polynomials

3.1 New Approximation Method for Sine Function Using Composition of the Minimax Approximate Polynomials

In [8], the error of the approximate comparison polynomial compared to the $\mathrm{comp}(a,b)$ is required to be bounded by $2^{-\alpha}$ for any $a, b \in [0,1]$ satisfying $|a-b| \geq \epsilon$. Note that $\mathrm{comp}(a,b) = \frac{\mathrm{sgn}(a-b)+1}{2}$. If a polynomial $p(x)$ approximating $\mathrm{sgn}(x)$ is $(\alpha-1,\epsilon)$-close, then the error of $\frac{p(a-b)+1}{2}$ compared to $\mathrm{comp}(a,b)$ is bounded by $2^{-\alpha}$ for any $a, b \in [0,1]$ satisfying $|a-b| \geq \epsilon$. Thus, we find composite polynomials approximating $\mathrm{sgn}(x)$ that satisfy $(\alpha-1,\epsilon)$-close to compare the performance of the proposed homomorphic comparison method fairly with that of the previous method in [8].

In [8], $\mathrm{sgn}(x)$ was approximated by using a composite polynomial whose component polynomial is f_n on $[-1,1]$, which satisfies the following three properties:

(i) $f_n(-x) = -f_n(x)$
(ii) $f_n(1) = 1, f_n(-1) = -1$
(iii) $f_n'(x) = c(1-x)^n(1+x)^n$ for some constant $c > 0$.

Then, the only polynomial satisfying the above three properties is given as

$$f_n(x) = \sum_{i=0}^{n} \frac{1}{4^i} \binom{2i}{i} x(1-x^2)^i.$$

If n and the number of compositions s_n of f_n become larger, the composite polynomial $f_n^{(s_n)}$ approximates $\mathrm{sgn}(x)$ better. In [8], it is stated that they have the best performance when $n = 4$. In addition, by defining and using the other polynomial g_n together with f_n for composition, the efficiency of the composite polynomial is further improved with the smaller number of the required compositions. However, the polynomial f_n that satisfies the above three properties does not guarantee the optimality for approximation using a composite polynomial. The other polynomial g_n defined in [8] has good properties, but it does not guarantee the optimality, too.

In this paper, we construct composite polynomials using new component polynomials f_i's, which are different from those used in the previous paper [8] and the repeated composition of each f_i is not used, that is, $s_i = 1$ for all i. Let $f_k \circ f_{k-1} \circ \cdots \circ f_1$ be a composite polynomial of component polynomials with odd degree terms approximating $\mathrm{sgn}(x)$ on $[-1,-\epsilon] \cup [\epsilon,1]$. Let $[a_0,b_0] = [\epsilon,1]$, $f_1([a_0,b_0]) = [a_1,b_1]$, $f_2([a_1,b_1]) = [a_2,b_2], \cdots, f_k([a_{k-1},b_{k-1}]) = [a_k,b_k]$. Note that $f_k \circ f_{k-1} \circ \cdots \circ f_1$ is $(\alpha-1,\epsilon)$-close if and only if $f_k \circ f_{k-1} \circ \cdots \circ f_1([\epsilon,1]) = [a_k,b_k] \subseteq [1-2^{1-\alpha}, 1+2^{1-\alpha}]$. Since $[a_k,b_k]$ should be a very small interval, it is desirable for each component polynomial f_i on the domain $[a_{i-1},b_{i-1}]$ to reduce the range as much as possible. Our key observation is that if the minimax

approximate polynomials are used as component polynomials, the size of the range $[a_i, b_i]$ can be reduced quickly as i increases. Thus, we use a composite polynomial of component minimax approximate polynomials obtained by the modified Remez algorithm.

In this paper, the Paterson-Stockmeyer algorithm [14] is used for evaluating the approximate polynomials. Table 1 shows the required depth consumption and the number of non-scalar multiplications for evaluating the approximate polynomials with odd degree terms using the Paterson-Stockmeyer algorithm. The exact required number of non-scalar multiplications and the depth consumption differ slightly depending on how the original Paterson-Stockmeyer algorithm [14] is modified. We refer to several papers and find the minimum number of required non-scalar multiplications and the depth consumption among them for each degree. We refer to the values in [8] for polynomials of degree smaller than or equal to 15 and the values in [11] for polynomials of degree larger than or equal to 17. The required depth consumption and the number of non-scalar multiplications for evaluating a polynomial of degree d with odd degree terms by using the Paterson-Stockmeyer algorithm are denoted by $\mathsf{dep}(d)$ and $\mathsf{mult}(d)$.

Table 1. The required depth consumption and the number of non-scalar multiplications for evaluating polynomials with odd degree terms using Paterson-Stockmeyer algorithm [8, 11]

polynomial degree d	$\mathsf{dep}(d)$	$\mathsf{mult}(d)$
3	2	2
5	3	3
7	3	4
9	4	4
11	4	5
13	4	6
15	4	7
17	5	7
19	5	8
21	5	8
23	5	8
25	5	10
27	5	10
29	5	10
31	5	10

The following definitions are necessary for description of Lemma 3.

Definition 5 ([8]). *For $\alpha > 0$ and $0 < \delta < 1$, a polynomial $p(x)$ is said to be (α, δ)-two-sided-close to* $\mathrm{sgn}(x)$ *if p satisfies the following:*

$$||p(x) - \mathrm{sgn}(x)||_{\infty,[-1-\delta,-1+\delta]\cup[1-\delta,1+\delta]} \leq 2^{-\alpha},$$

where $||\cdot||_{\infty,D}$ *denotes the infinity norm over the domain* D.

Definition 6. *Let* $\{f_i\}_{1\le i\le k}$ *be a set of polynomials satisfying* $\deg(f_i) = d_i$, $1 \le i \le k$. $\mathsf{MultNum}(\{f_i\}_{1\le i\le k})$ *and* $\mathsf{DepNum}(\{f_i\}_{1\le i\le k})$ *denote the sum of the numbers of non-scalar multiplications and the sum of depth consumptions required to evaluate* f_i *for* $1 \le i \le k$ *by using Paterson-Stockmeyer algorithm, respectively. That is,*

$$\mathsf{MultNum}(\{f_i\}_{1\le i\le k}) = \sum_{i=1}^{k} \mathsf{mult}(\deg(f_i))$$

$$\mathsf{DepNum}(\{f_i\}_{1\le i\le k}) = \sum_{i=1}^{k} \mathsf{dep}(\deg(f_i)).$$

Our goal is to find a $(\alpha-1,\epsilon)$-close composite polynomial $f_k \circ f_{k-1} \circ \cdots \circ f_1$ while minimizing $\mathsf{MultNum}(\{f_i\}_{1\le i\le k})$ and $\mathsf{DepNum}(\{f_i\}_{1\le i\le k})$. The following lemma implies that finding a $(\alpha-1,\epsilon)$-close composite polynomial $f_k \circ f_{k-1} \circ \cdots \circ f_1$ is equivalent to finding a $(\alpha-1,\delta)$-two-sided-close composite polynomial $\tilde{f}_k \circ \tilde{f}_{k-1} \circ \cdots \circ \tilde{f}_1$ when $\delta = \frac{1-\epsilon}{1+\epsilon}$.

Lemma 3. *For a set of polynomials with odd degree terms* $\{f_i\}_{1\le i\le k}$, *let* $\{\tilde{f}_i\}_{1\le i\le k}$ *be a set of polynomials with odd degree terms such that* $\tilde{f}_1(x) = f_1(\frac{1+\epsilon}{2}x)$ *and* $\tilde{f}_i(x) = f_i(x)$, $2 \le i \le k$. *Then,* $f_k \circ f_{k-1} \circ \cdots \circ f_1$ *is* $(\alpha-1,\epsilon)$*-close if and only if* $\tilde{f}_k \circ \tilde{f}_{k-1} \circ \cdots \circ \tilde{f}_1$ *is* $(\alpha-1,\delta)$*-two-sided-close when* $\delta = \frac{1-\epsilon}{1+\epsilon}$.

Proof. Let $f_k \circ f_{k-1} \circ \cdots \circ f_1$ be a $(\alpha-1,\epsilon)$-close composite polynomial of component polynomials with odd degree terms. Since $f_k \circ f_{k-1} \circ \cdots \circ f_1(x)$ is a polynomial with odd degree terms, it is sufficient to consider only when $x > 0$. Then, $f_k \circ f_{k-1} \circ \cdots \circ f_1(x) \in [1-2^{-(\alpha-1)}, 1+2^{-(\alpha-1)}]$ for $\epsilon \le x \le 1$. Let $x' = \frac{2}{1+\epsilon}x$. $\epsilon \le x \le 1$ corresponds to $1-\delta \le x' \le 1+\delta$. Then, $\tilde{f}_k \circ \tilde{f}_{k-1} \circ \cdots \circ \tilde{f}_1(x') = f_k \circ f_{k-1} \circ \cdots \circ f_1(x) \in [1-2^{-(\alpha-1)}, 1+2^{-(\alpha-1)}]$ for $1-\delta \le x' \le 1+\delta$. Thus, $\tilde{f}_k \circ \tilde{f}_{k-1} \circ \cdots \circ \tilde{f}_1$ is $(\alpha-1,\delta)$-two-sided-close. Conversely, let $\tilde{f}_k \circ \tilde{f}_{k-1} \circ \cdots \circ \tilde{f}_1(x') \in [1-2^{-(\alpha-1)}, 1+2^{-(\alpha-1)}]$ for $1-\delta \le x' \le 1+\delta$. Let $x = \frac{1+\epsilon}{2}x'$. $1-\delta \le x' \le 1+\delta$ corresponds to $\epsilon \le x \le 1$. Then, $f_k \circ f_{k-1} \circ \cdots \circ f_1(x) = \tilde{f}_k \circ \tilde{f}_{k-1} \circ \cdots \circ \tilde{f}_1(x') \in [1-2^{-(\alpha-1)}, 1+2^{-(\alpha-1)}]$ for $\epsilon \le x \le 1$, which means that $f_k \circ f_{k-1} \circ \cdots \circ f_1$ is $(\alpha-1,\epsilon)$-close. Thus, the lemma is proved. □

Note that since $\deg(f_i) = \deg(\tilde{f}_i), 1 \le i \le k$ in Lemma 3, it holds that

$$\mathsf{MultNum}(\{f_i\}_{1\le i\le k}) = \mathsf{MultNum}(\{\tilde{f}_i\}_{1\le i\le k})$$
$$\mathsf{DepNum}(\{f_i\}_{1\le i\le k}) = \mathsf{DepNum}(\{\tilde{f}_i\}_{1\le i\le k}).$$

Thus, for any $m, n \in \mathbb{N}$, a composite polynomial of component polynomials with odd degree terms $f_k \circ f_{k-1} \circ \cdots \circ f_1$ is $(\alpha-1,\epsilon)$-close and satisfies

$\mathsf{MultNum}(\{f_i\}_{1\le i\le k}) = m$ and $\mathsf{DepNum}(\{f_i\}_{1\le i\le k}) = n$ if and only if the corresponding composite polynomial $\tilde{f}_k \circ \tilde{f}_{k-1} \circ \cdots \circ \tilde{f}_1$ is $(\alpha-1, \delta)$-two-sided-close and satisfies $\mathsf{MultNum}(\{\tilde{f}_i\}_{1\le i\le k}) = m$ and $\mathsf{DepNum}(\{\tilde{f}_i\}_{1\le i\le k}) = n$ when $\delta = \frac{1-\epsilon}{1+\epsilon}$. Thus, it can be seen that the following two algorithms are equivalent:

(i) An algorithm that finds the $(\alpha-1, \epsilon)$-close composite polynomial $f_k \circ \cdots \circ f_1$ which minimizes the number of non-scalar multiplications and the depth consumption
(ii) An algorithm that finds the $(\alpha-1, \delta)$-two-sided-close composite polynomial $\tilde{f}_k \circ \tilde{f}_{k-1} \circ \cdots \circ \tilde{f}_1$ which minimizes the number of non-scalar multiplications and the depth consumption

Thus, from now on, we focus on finding $(\alpha-1, \delta)$-two-sided-close composite polynomial $\tilde{f}_k \circ \tilde{f}_{k-1} \circ \cdots \circ \tilde{f}_1$ which minimizes the number of non-scalar multiplications and the depth consumption.

The minimax composite polynomial, which is the core of the proposed homomorphic comparison method, is now defined as follows. The main idea of the proposed approximation method is to use the minimax composite polynomial to approximate the sign function. We denote $[-1-s, -1+s] \cup [1-s, 1+s]$ by R_s for $s > 0$.

Definition 7. *Let $\{f_i\}_{1\le i\le k}$ be a set of polynomials. Let D be $[-b,-a] \cup [a,b]$. $f_k \circ f_{k-1} \circ \cdots \circ f_1$ is called a minimax composite polynomial on D if there exists $\{d_i\}_{1\le i\le k}$ that satisfies the followings:*

- *f_1 is the minimax approximate polynomial of degree at most d_1 on D for $\mathrm{sgn}(x)$ and the minimax approximation error is equal to τ_1.*
- *For $2 \le i \le k$, f_i is the minimax approximate polynomial of degree at most d_i on $f_{i-1} \circ f_{i-2} \circ \cdots \circ f_1(D)$ for $\mathrm{sgn}(x)$. The minimax approximation error is τ_i.*

Note that $f_i \circ f_{i-1} \circ \cdots \circ f_1(D) = R_{\tau_i}$, $1 \le i \le k$ from Theorem 1. In fact, τ_i becomes smaller as i increases. It can be seen that if $f_k \circ f_{k-1} \circ \cdots \circ f_1$ is a minimax composite polynomial on $D = [-b,-a] \cup [a,b]$, then $\{f_i\}_{1\le i\le k}$ is a set of polynomials with odd degree terms from Lemma 2. If $\tau_k \le 2^{-(\alpha-1)}$, then the minimax composite polynomial on R_δ becomes $(\alpha-1, \delta)$-two-sided-close. Our key idea is to find the minimax composite polynomial on R_δ that requires the minimum number of non-scalar multiplications and depth consumption among all $(\alpha-1, \delta)$-two-sided-close minimax composite polynomials on R_δ. Note that there is a tradeoff between the number of non-scalar multiplications and the depth consumption. We deal with both cases when putting priority on minimizing the number of non-scalar multiplications and on minimizing the depth consumption.

3.2 Optimality of Approximation of the Sign Function by a Minimax Composite Polynomial

Since $\mathrm{sgn}(x)$ is an odd function, it is natural to approximate $\mathrm{sgn}(x)$ by using a composite polynomial of component polynomials with odd degree terms. Assume

that we can obtain the minimax composite polynomial on R_δ that requires the minimum number of non-scalar multiplications and depth consumption among all minimax composite polynomials on R_δ satisfying $(\alpha-1,\delta)$-two-sided-close. In this subsection, it is proved that the obtained minimax composite polynomial on R_δ requires less number of non-scalar multiplications and depth consumption than any $(\alpha-1,\delta)$-two-sided-close composite polynomial of component polynomials with odd degree terms. That is, for any $(\alpha-1,\delta)$-two-sided-close composite polynomial of component polynomials with odd degree terms, there exists a $(\alpha-1,\delta)$-two-sided-close minimax composite polynomial on R_δ such that the number of required non-scalar multiplications and the depth consumption for the minimax composite polynomial are less than or equal to those for the composite polynomial of component polynomials with odd degree terms, respectively.

The following definition and lemmas are needed for the proof of optimality of the proposed approximation method of approximating the sign function using a minimax composite polynomial.

Definition 8. *Let $\{f_i\}_{1\le i\le k}$ be a set of polynomials. $f_k\circ f_{k-1}\circ\cdots\circ f_1$ is called a 1-centered range composite polynomial on R_δ if $\{f_i\}_{1\le i\le k}$ is a set of polynomials with odd degree terms and there exists $\{\tau_i\}_{1\le i\le k}$ such that $f_1([1-\delta,1+\delta]) = [1-\tau_1,1+\tau_1]$ and $f_i([1-\tau_{i-1},1+\tau_{i-1}]) = [1-\tau_i,1+\tau_i]$ for $2\le i\le k$.*

Lemma 4. *Let f_1 be the minimax approximate polynomial of degree at most d on $[-b_1,-a_1]\cup[a_1,b_1]$ for $\mathrm{sgn}(x)$. Let f_2 be the minimax approximate polynomial of degree at most d on $[-b_2,-a_2]\cup[a_2,b_2]$ for $\mathrm{sgn}(x)$. If $[a_2,b_2]\subseteq[a_1,b_1]$, then the minimax approximation error e_2 of f_2 is less than or equal to the minimax approximation error e_1 of f_1.*

Proof. When f_1 approximates $\mathrm{sgn}(x)$ on $[-b_1,-a_1]\cup[a_1,b_1]$, the maximum approximation error e_1 is larger than or equal to the maximum approximation error e_1' when f_1 approximates $\mathrm{sgn}(x)$ on $[-b_2,-a_2]\cup[a_2,b_2]$. According to the definition of minimax approximate polynomial, f_2 is the polynomial with the smallest maximum approximation error when approximating $\mathrm{sgn}(x)$ on $[-b_2,-a_2]\cup[a_2,b_2]$ among all polynomials of degree smaller than or equal to d. Among polynomials of degree smaller than or equal to d, there is also f_1. Thus, it holds that $e_2\le e_1'\le e_1$, and the lemma is proved. □

Lemma 5. *Let $\tilde{f}_k\circ\tilde{f}_{k-1}\circ\cdots\circ\tilde{f}_1$ be any $(\alpha-1,\delta)$-two-sided-close 1-centered range composite polynomial on R_δ. Then, there is a $(\alpha-1,\delta)$-two-sided-close minimax composite polynomial $\hat{f}_k\circ\hat{f}_{k-1}\circ\cdots\circ\hat{f}_1$ on R_δ such that $\deg(\hat{f}_i)\le\deg(\tilde{f}_i)$ for i, $1\le i\le k$.*

Proof. Since $\tilde{f}_k\circ\tilde{f}_{k-1}\circ\cdots\circ\tilde{f}_1$ is a 1-centered range composite polynomial on R_δ, there exists $\{\tau_i\}_{1\le i\le k}$ such that $\tilde{f}_1([1-\delta,1+\delta]) = [1-\tau_1,1+\tau_1]$ and $\tilde{f}_i([1-\tau_{i-1},1+\tau_{i-1}) = [1-\tau_i,1+\tau_i]$ for all i, $2\le i\le k$. Then, $\tilde{f}_k\circ\tilde{f}_{k-1}\circ\cdots\circ\tilde{f}_1([1-\delta,1+\delta]) = [1-\tau_k,1+\tau_k]$. Since $\tilde{f}_k\circ\tilde{f}_{k-1}\circ\cdots\circ\tilde{f}_1$ is $(\alpha-1,\delta)$-two-sided-close, $\tau_k\le 2^{-(\alpha-1)}$ should hold. Let $\deg(\tilde{f}_i)=d_i$, $1\le i\le k$. Let $\hat{f}_1$ be the minimax approximate polynomial of degree at most d_1 on R_δ and let τ_1' be the

approximation error of $\hat{f}_1$. Then $\tau_1' \leq \tau_1$. Let τ_i' be the approximation error of $\hat{f}_i$, which is the minimax approximate polynomial of degree at most d_i on $R_{\tau_{i-1}'}$ for $\mathrm{sgn}(x)$ for $i, 2 \leq i \leq k$. Then, $\hat{f}_k \circ \hat{f}_{k-1} \circ \cdots \circ \hat{f}_1$ is a minimax composite polynomial on R_δ. We want to show that $\tau_i' \leq \tau_i$, $2 \leq i \leq k$ by inductive method. Assume that $\tau_{i-1}' \leq \tau_{i-1}$. Let τ_i'' be the approximation error of the minimax approximate polynomial of degree at most d_i on $R_{\tau_{i-1}}$ for $\mathrm{sgn}(x)$. From Lemma 4, it holds that $\tau_i' \leq \tau_i''$. Since $\tilde{f}_i([1-\tau_{i-1}, 1+\tau_{i-1}]) = [1-\tau_i, 1+\tau_i]$, $\tau_i'' \leq \tau_i$ holds. Thus, $\tau_i' \leq \tau_i'' \leq \tau_i$. It holds that $\tau_i' \leq \tau_i$ for all i, $2 \leq i \leq k$ by inductive method. Since $\tau_k' \leq \tau_k \leq 2^{-(\alpha-1)}$, $\hat{f}_k \circ \hat{f}_{k-1} \circ \cdots \circ \hat{f}_1$ is a $(\alpha-1, \delta)$-two-sided-close minimax composite polynomial on R_δ such that $\deg(\hat{f}_i) \leq \deg(\tilde{f}_i)$ for all i, $1 \leq i \leq k$. □

Lemma 6. *Let $f_k \circ f_{k-1} \circ \cdots \circ f_1$ be any $(\alpha-1, \delta)$-two-sided-close composite polynomial of component polynomials with odd degree terms. Then, there is a $(\alpha-1, \delta)$-two-sided-close 1-centered range composite polynomial $\tilde{f}_k \circ \tilde{f}_{k-1} \circ \cdots \circ \tilde{f}_1$ on R_δ such that $\deg(\tilde{f}_i) = \deg(f_i)$ for all i, $1 \leq i \leq k$.*

Proof. Let $f_1([1-\delta, 1+\delta]) = [a_1, b_1]$, $f_2([a_1, b_1]) = [a_2, b_2], \cdots, f_k([a_{k-1}, b_{k-1}]) = [a_k, b_k]$. Since $\{f_i\}_{1 \leq i \leq k}$ is a set of polynomials with odd degree terms, it holds that $f_1([-1-\delta, -1+\delta]) = [-b_1, -a_1], f_2([-b_1, -a_1]) = [-b_2, -a_2], \cdots, f_k([-b_{k-1}, -a_{k-1}]) = [-b_k, -a_k]$. Satisfying $(\alpha-1, \delta)$-two-sided-close means that $[a_k, b_k] \subseteq [1-2^{-(\alpha-1)}, 1+2^{-(\alpha-1)}]$. Also, it is easy to see that $0 < a_i < b_i$ for $i, 1 \leq i \leq k$ from the fact that $f_k \circ f_{k-1} \circ \cdots \circ f_1$ is a $(\alpha-1, \delta)$-two-sided-close composite polynomial. Let $\tilde{f}_1(x) = \frac{2}{a_1+b_1} f_1(x)$ and $\tilde{f}_i(x) = \frac{2}{a_i+b_i} f_i(\frac{a_i+b_i}{2}x)$, $2 \leq i \leq k$. Then, $\tilde{f}_1([1-\delta, 1+\delta]) = [1-\frac{b_1-a_1}{a_1+b_1}, 1+\frac{b_1-a_1}{a_1+b_1}]$ and $\tilde{f}_i([1-\frac{b_{i-1}-a_{i-1}}{a_{i-1}+b_{i-1}}, 1+\frac{b_{i-1}-a_{i-1}}{a_{i-1}+b_{i-1}}]) = [1-\frac{b_i-a_i}{a_i+b_i}, 1+\frac{b_i-a_i}{a_i+b_i}]$, $2 \leq i \leq k$. Then, $\tilde{f}_k \circ \tilde{f}_{k-1} \circ \cdots \circ \tilde{f}_1$ is a 1-centered range composite polynomial on R_δ. We want to show that $[1-\frac{b_k-a_k}{a_k+b_k}, 1+\frac{b_k-a_k}{a_k+b_k}] = [\frac{2a_k}{a_k+b_k}, \frac{2b_k}{a_k+b_k}] \subseteq [1-2^{-(\alpha-1)}, 1+2^{-(\alpha-1)}]$, which means that $\tilde{f}_k \circ \tilde{f}_{k-1} \circ \cdots \circ \tilde{f}_1$ is $(\alpha-1, \delta)$-two-sided-close. If $a_k + b_k \leq 2$, then $1-2^{-(\alpha-1)} \leq a_k \leq \frac{2a_k}{a_k+b_k}$ and $\frac{2b_k}{a_k+b_k} = 2 - \frac{2a_k}{a_k+b_k} \leq 2 - a_k \leq 1+2^{-(\alpha-1)}$ hold. On the other hand, if $a_k + b_k > 2$, then $\frac{2b_k}{a_k+b_k} < b_k \leq 1+2^{-(\alpha-1)}$ and $\frac{2a_k}{a_k+b_k} = 2 - \frac{2b_k}{a_k+b_k} > 2 - b_k \geq 1-2^{-(\alpha-1)}$ hold. Thus $\tilde{f}_k \circ \tilde{f}_{k-1} \circ \cdots \circ \tilde{f}_1$ is a $(\alpha-1, \delta)$-two-sided-close 1-centered range composite polynomial on R_δ satisfying $\deg(\tilde{f}_i) = \deg(f_i)$ for all i, $1 \leq i \leq k$. □

The following procedure for proof of Theorem 2 is used:

(i) It is proved that for any $(\alpha-1, \delta)$-two-sided-close 1-centered range composite polynomial $\tilde{f}_k \circ \tilde{f}_{k-1} \circ \cdots \circ \tilde{f}_1$ on R_δ, it holds that $\deg(\hat{f}_i) = \deg(\tilde{f}_i)$ for all i, $1 \leq i \leq k$ for some $(\alpha-1, \delta)$-two-sided-close minimax composite polynomial $\hat{f}_k \circ \hat{f}_{k-1} \circ \cdots \circ \hat{f}_1$ on R_δ from Lemma 5.

(ii) It is proved that for any $(\alpha-1, \delta)$-two-sided-close composite polynomial of component polynomials with odd degree terms $\{f_i\}_{1 \leq i \leq k}$, it holds that $\deg(\tilde{f}_i) \leq \deg(f_i)$ for all i, $1 \leq i \leq k$ for some $(\alpha-1, \delta)$-two-sided-close 1-centered range composite polynomial $\tilde{f}_k \circ \tilde{f}_{k-1} \circ \cdots \circ \tilde{f}_1$ on R_δ from Lemma 6.

(iii) Finally, with above lemmas, it is proved in Theorem 2 that for any $(\alpha-1,\delta)$-two-sided-close composite polynomial of component polynomials with odd degree terms $\{f_i\}_{1\le i\le k}$, it holds that $\deg(\hat{f}_i) \le \deg(f_i)$ for all i, $1 \le i \le k$ for some $(\alpha-1,\delta)$-two-sided-close minimax composite polynomial $\hat{f}_k \circ \hat{f}_{k-1} \circ \cdots \circ \hat{f}_1$ on R_δ.

Theorem 2. *Let $f_k \circ f_{k-1} \circ \cdots \circ f_1$ be any $(\alpha-1,\delta)$-two-sided-close composite polynomial of component polynomials with odd degree terms. Then, there is a $(\alpha-1,\delta)$-two-sided-close minimax composite polynomial $\hat{f}_k \circ \hat{f}_{k-1} \circ \cdots \circ \hat{f}_1$ on R_δ such that $\deg(\hat{f}_i) \le \deg(f_i)$ for all i, $1 \le i \le k$.*

Proof. Let $f_k \circ f_{k-1} \circ \cdots \circ f_1$ be any $(\alpha-1,\delta)$-two-sided-close composite polynomial of component polynomials with odd degree terms. From Lemma 6, there is a $(\alpha-1,\delta)$-two-sided-close 1-centered range composite polynomial $\tilde{f}_k \circ \tilde{f}_{k-1} \circ \cdots \circ \tilde{f}_1$ on R_δ such that $\deg(\tilde{f}_i) = \deg(f_i)$, $1 \le i \le k$. In addition, from Lemma 5, there is a $(\alpha-1,\delta)$-two-sided-close minimax composite polynomial $\hat{f}_k \circ \hat{f}_{k-1} \circ \cdots \circ \hat{f}_1$ on R_δ such that $\deg(\hat{f}_i) \le \deg(\tilde{f}_i)$, $1 \le i \le k$. Thus, there is a $(\alpha-1,\delta)$-two-sided-close minimax composite polynomial $\hat{f}_k \circ \hat{f}_{k-1} \circ \cdots \circ \hat{f}_1$ on R_δ such that $\deg(\hat{f}_i) \le \deg(f_i)$ for all i, $1 \le i \le k$. □

Remark 2. In Theorem 2, since $\deg(\hat{f}_i) \le \deg(f_i)$ for $1 \le i \le k$, it holds that $\mathsf{MultNum}(\{\hat{f}_i\}_{1\le i\le k}) \le \mathsf{MultNum}(\{f_i\}_{1\le i\le k})$ and $\mathsf{DepNum}(\{\hat{f}_i\}_{1\le i\le k}) \le \mathsf{DepNum}(\{f_i\}_{1\le i\le k})$. It means that if we find the minimax composite polynomial on R_δ that requires the minimum number of non-scalar multiplications and depth consumption among all $(\alpha-1,\delta)$-two-sided-close minimax composite polynomials on R_δ, the number of required non-scalar multiplications and depth consumption for the obtained minimax composite polynomial on R_δ are less than or equal to those for any $(\alpha-1,\delta)$-two-sided-close composite polynomial of component polynomials with odd degree terms, respectively.

3.3 Achieving Polynomial-Time Algorithm for New Approximation Method by Using Dynamic Programming

We can find the $(\alpha-1,\delta)$-two-sided-close minimax composite polynomial on R_δ that requires the minimum number of non-scalar multiplications and depth consumption among all $(\alpha-1,\delta)$-two-sided-close minimax composite polynomials by brute-force search. However, the brute-force search requires considerable time. Thus, dynamic programming is used to find the minimax composite polynomial on R_δ with the computational complexity in polynomial time. Thus, we propose an algorithm to find the minimax composite polynomial on R_δ that requires the minimum number of non-scalar multiplications and depth consumption in polynomial time by using dynamic programming.

$\mathsf{MinErr}(d,t)$, $\mathsf{InvMinErr}(d,t)$, $f(m,n,t)$, and $G(m,n,t)$ are defined before the description of the proposed algorithms as follows.

Definition 9. *For $d \in \mathbb{N}$ and $t \in (0,1)$, $\mathsf{MinErr}(d,t)$ is the minimax approximation error of the minimax approximate polynomial of degree at most d on R_t for* $\mathrm{sgn}(x)$.

Lemma 7. *For a fixed odd $d \in \mathbb{N}$, $\mathsf{MinErr}(d,t)$ is a strictly increasing continuous function of t on $(0,1)$.*

Proof. Let d be $2i+1$. Consider the minimax approximate polynomial $p(x)$ of degree at most $2i+1$ on R_t for $\mathrm{sgn}(x)$. Let τ_0 be the minimax approximaton error of $p(x)$ on R_t. Since $\mathrm{sgn}(x)$ is an odd function, it can be seen from Lemma 2 that the minimax approximate polynomial of degree at most $2i+1$ to $\mathrm{sgn}(x)$ is equal to the minimax approximate polynomial of degree at most $2i+2$ to $\mathrm{sgn}(x)$. Also, $p(x)$ is a polynomial with odd degree terms from Lemma 2. We want to show that there exist $i+2$ distinct points $x_0, x_1, \cdots, x_{i+1} \in [1-t, 1+t]$ that satisfy the following three properties:

Prop 1. $1-t = x_0 < x_1 < \cdots < x_{i+1} = 1+t$.
Prop 2. $p(x_j) = 1 + (-1)^{j+1}\tau_0$, $0 \le j \le i+1$.
Prop 3. $p(x)$ is strictly increasing on $(0, x_1)$. For j, $1 \le j \le i$, $p(x)$ is strictly increasing on (x_j, x_{j+1}) when j is even, and strictly decreasing on (x_j, x_{j+1}) when j is odd. Also, $p(x)$ is strictly increasing on (x_i, ∞) if i is even and strictly decreasing on (x_i, ∞) if i is odd.

$|p(x) - \mathrm{sgn}(x)|$ should have maximum values at $2i+4$ distinct points in R_t from Theorem 1. However, there are at most $2i$ distinct points x such that $p'(x) = 0$. If we consider when $x > 0$, $|p(x) - \mathrm{sgn}(x)|$ should have maximum values at $i+2$ distinct points on $[1-t, 1+t]$ and there are at most i distinct points x such that $p'(x) = 0$. If $|p(x) - \mathrm{sgn}(x)|$ has maximum value at $x = x_0$, then it holds that $p'(x_0) = 0$ or $x = x_0$ is a boundary point, that is, $x_0 \in \{1-t, 1+t\}$. Thus, $|p(x) - \mathrm{sgn}(x)|$ should have maximum values at two boundary points $x = 1-t$ and $x = 1+t$. Let $x_0, \cdots, x_{i+1}$ be the $i+2$ distinct points on $(0, \infty)$ such that $|p(x) - \mathrm{sgn}(x)|$ has maximum values at those points. Then, it holds that $x_0 = 1-t, x_{i+1} = 1+t$ and $p'(x_1) = p'(x_2) = \cdots, p'(x_i) = 0$. Also, considering $p(0) = 0$ and $p(x_1) > 0$, $p(x)$ is strictly increasing on $(0, x_1)$. Since $p(x_0) < p(x_1)$, it holds that $p(x_0) = 1 - \tau_0, p(x_1) = 1 + \tau_0, p(x_2) = 1 - \tau_0, \cdots$ from Theorem 1. Also, it can be seen that the Prop 3 is satisfied from Theorem 1. Thus, there exist $i+2$ points $x_0, x_1, \cdots, x_{i+1} \in (0, \infty)$ that satisfy the above three properties. Now we want to show that $\mathsf{MinErr}(d,t)$ is a strictly increasing continuous function of t with domain $(0,1)$ as follows:

(i) Strictly increasing:
Let $0 < t_1 < t_2 < 1$. Let $p_1(x)$ and $p_2(x)$ be the minimax approximate polynomials of degree at most $2i+1$ on R_{t_1} and R_{t_2}, respectively. It is trivial that $\mathsf{MinErr}(d,t_1) \le \mathsf{MinErr}(d,t_2)$. Assume that $\mathsf{MinErr}(d,t_1) = \mathsf{MinErr}(d,t_2) = \tau_0$. Then, by the uniqueness property of the minimax approximate polynomial, it should be hold that $p_1(x) = p_2(x)$. Note that $p_1(x)$ is the minimax approximate polynomial of degree at most $2i+1$ on R_{t_1}. Then, it can be

seen that $0 < p_1(1-t_2) < p_1(1-t_1) = 1-\tau_0$ from Prop 3. Considering $p_1(x) = p_2(x)$, the minimax approximation error of $p_2(x)$ on R_{t_2} is larger than τ_0. That is, $\mathsf{MinErr}(d,t_1) < \mathsf{MinErr}(d,t_2)$, which is a contradiction. Thus, $\mathsf{MinErr}(d,t)$ is a strictly increasing function of t.

(ii) Continuous:
We want to show that $\mathsf{MinErr}(d,t)$ is continuous at $t=t_0$, that is, for any $\delta' > 0$, there exists $\epsilon' > 0$ such that $|t-t_0| \le \epsilon'$ implies $|\mathsf{MinErr}(d,t) - \mathsf{MinErr}(d,t_0)| \le \delta'$. Let $p(x)$ be the minimax approximate polynomial of degree at most $2i+1$ on R_{t_0}, and let τ_0 be the minimax approximation error of $p(x)$. It is enough to consider only the case when $\delta' < \tau_0$. There exist $i+2$ distinct points $x_0, x_1, \cdots, x_{i+1} \in (0,\infty)$ that satisfy the above three properties. There exists a unique $x \in (0, x_0)$ such that $p(x) = 1-\tau_0-\delta'$. Let ϵ'_1 be $1-t_0-x$ for the unique x. Also, there exists unique $x \in (x_{i+1}, \infty)$ such that $p(x) = 1+\tau_0+\delta'$ when i is even and unique $x \in (x_{i+1}, \infty)$ such that $p(x) = 1-\tau_0-\delta'$ when i is odd. Let ϵ'_2 be $x-1-t_0$ for the unique x. There exists unique $x \in (x_0, x_1)$ such that $p(x) = 1-\tau_0+\delta'$. Let ϵ'_3 be $x-1+t_0$ for the unique x. Also, there exists unique $x \in (x_i, x_{i+1})$ such that $p(x) = 1+\tau_0-\delta'$ when i is even and unique $x \in (x_i, x_{i+1})$ such that $p(x) = 1-\tau_0+\delta'$. Let ϵ'_4 be $-x+1+t_0$ for the unique x. Now, let ϵ' be $\min(\epsilon'_1, \epsilon'_2, \epsilon'_3, \epsilon'_4)$. Then, $p([1-t_0-\epsilon', 1+t_0+\epsilon']) \subseteq [1-\tau_0-\delta', 1+\tau_0+\delta']$. Thus, the minimax approximation error of the minimax approximate polynomial on $R_{t_0+\epsilon'}$ is smaller than or equal to $\tau_0+\delta'$. That is, $\mathsf{MinErr}(d,t_0+\epsilon') \le \tau_0+\delta'$. On the other hand, let $x'_0 = x_0+\epsilon', x'_1 = x_1, \cdots, x'_i = x_i, x'_{i+1} = x_{i+1}-\epsilon'$. Consider $2i+4$ points $-x'_{i+1}, -x'_i, \cdots, -x'_0, x'_0, \cdots, x'_i, x'_{i+1}$. From Lemma 1, the minimax approximation error of the minimax approximate polynomial on $R_{t_0-\epsilon'}$ is larger than or equal to $\tau_0-\delta'$. That is, $\mathsf{MinErr}(d,t_0-\epsilon') \ge \tau_0-\delta'$. Since $\mathsf{MinErr}(d,t)$ is an increasing function, if $|t-t_0| \le \epsilon'$, then $|\mathsf{MinErr}(d,t) - \mathsf{MinErr}(d,t_0)| \le \delta'$. Thus, $\mathsf{MinErr}(d,t)$ is continuous at $t=t_0$.

□

If the minimax approximate polynomial of degree at most d on R_t narrows the domain R_t to a range R_τ, $\mathsf{MinErr}(d,t)$ outputs τ. Since $\mathsf{MinErr}(d,t)$ is strictly increasing function of t on $(0,\infty)$, the inverse function of $\mathsf{MinErr}(d,t)$ exists, which is defined as follows.

Definition 10. *For $d \in \mathbb{N}$, $\mathsf{InvMinErr}(d,t)$ is $\tau > 0$ such that $\mathsf{MinErr}(d,\tau) = t$.*

The approximate value of $\mathsf{InvMinErr}(d,t)$ can be obtained by binary search using modified Remez algorithm.

Definition 11. *$f(m,n,t)$ is the maximum $\tau \in (0,1)$ such that there exists a minimax composite polynomial $f_k \circ f_{k-1} \circ \cdots \circ f_1$ on R_τ satisfying $f_k \circ f_{k-1} \circ \cdots \circ f_1([1-\tau, 1+\tau]) \subseteq [1-t, 1+t]$, $\mathsf{MultNum}(\{f_i\}_{1\le i\le k}) \le m$, and $\mathsf{DepNum}(\{f_i\}_{1\le i\le k}) \le n$.*

$f(m,n,t)$ outputs the maximum $\tau > 0$ when the range of a minimax composite polynomial on R_τ becomes smaller than R_t with m or less number of

non-scalar multiplications and with n or less depth consumption. The degrees of k component polynomials for the corresponding minimax composite polynomial $f_k \circ f_{k-1} \circ \cdots \circ f_1$ on R_τ in Definition 11 are stored in $G(m, n, t)$ as an ordered set. It is trivial that if $0 \leq m \leq 1$ or $0 \leq n \leq 1$, then $f(m, n, t) = t$. For $m \geq 2$ and $n \geq 2$, the following theorem for $f(m, n, t)$ holds:

Theorem 3. *For $m \geq 2$ and $n \geq 2$, the following recursion for $f(m, n, t)$ holds:*

$$f(m,n,t) = \max_{\substack{1\leq k \\ \mathsf{mult}(2k+1)\leq m \\ \mathsf{dep}(2k+1)\leq n}} \mathsf{InvMinErr}(2k+1, f(m-\mathsf{mult}(2k+1), n-\mathsf{dep}(2k+1), t)).$$

Proof. Let $\tau = f(m, n, t)$. Assume that

$$\tau > \max_{\substack{1\leq k \\ \mathsf{mult}(2k+1)\leq m \\ \mathsf{dep}(2k+1)\leq n}} \mathsf{InvMinErr}(2k+1, f(m-\mathsf{mult}(2k+1), n-\mathsf{dep}(2k+1), t)).$$

Then there exists a minimax composite polynomial $f_k \circ f_{k-1} \circ \cdots \circ f_1$ satisfying $f_k \circ f_{k-1} \circ \cdots \circ f_1([1-\tau, 1+\tau]) \subseteq [1-t, 1+t], \mathsf{MultNum}(\{f_i\}_{1\leq i\leq k}) \leq m$, and $\mathsf{DepNum}(\{f_i\}_{1\leq i\leq k}) \leq n$. Let d_1 be the degree of f_1 and let $f_1([1-\tau, 1+\tau]) = [1-\tau', 1+\tau']$. Since the minimax composite polynomial $f_k \circ f_{k-1} \circ \cdots \circ f_2$ on $[1-\tau', 1+\tau']$ satisfies $f_k \circ f_{k-1} \circ \cdots \circ f_2([1-\tau', 1+\tau']) \subseteq [1-t, 1+t]$, $\mathsf{MultNum}(\{f_i\}_{2\leq i\leq k}) \leq m - \mathsf{mult}(d_1)$, and $\mathsf{DepNum}(\{f_i\}_{2\leq i\leq k}) \leq n - \mathsf{dep}(d_1)$, it holds that $\tau' \leq f(m-\mathsf{mult}(d_1), n-\mathsf{dep}(d_1), t)$. Then,

$$\begin{aligned}\tau = \mathsf{InvMinErr}(d_1, \tau') &\leq \mathsf{InvMinErr}(d_1, f(m-\mathsf{mult}(d_1), n-\mathsf{dep}(d_1), t)) \\ &\leq \max_{\substack{1\leq k \\ \mathsf{mult}(2k+1)\leq m \\ \mathsf{dep}(2k+1)\leq n}} \mathsf{InvMinErr}(2k+1, f(m-\mathsf{mult}(2k+1), n-\mathsf{dep}(2k+1), t))\end{aligned}$$

This leads to a contradiction because

$$\tau > \max_{\substack{1\leq k \\ \mathsf{mult}(2k+1)\leq m \\ \mathsf{dep}(2k+1)\leq n}} \mathsf{InvMinErr}(2k+1, f(m-\mathsf{mult}(2k+1), n-\mathsf{dep}(2k+1), t)).$$

Assume that

$$\tau < \max_{\substack{1\leq k \\ \mathsf{mult}(2k+1)\leq m \\ \mathsf{dep}(2k+1)\leq n}} \mathsf{InvMinErr}(2k+1, f(m-\mathsf{mult}(2k+1), n-\mathsf{dep}(2k+1), t)).$$

$$\max_{\substack{1\leq k \\ \mathsf{mult}(2k+1)\leq m \\ \mathsf{dep}(2k+1)\leq n}} \mathsf{InvMinErr}(2k+1, f(m-\mathsf{mult}(2k+1), n-\mathsf{dep}(2k+1), t)) =$$

$\mathsf{InvMinErr}(2i+1, f(m-\mathsf{mult}(2i+1), n-\mathsf{dep}(2i+1), t))$ for some i. Let τ' be

$\mathsf{InvMinErr}(2i+1, f(m-\mathsf{mult}(2i+1), n-\mathsf{dep}(2i+1), t))$. Let τ'' be $f(m-\mathsf{mult}(2i+1), n-\mathsf{dep}(2i+1), t) = \mathsf{MinErr}(2i+1, \tau')$. Then, there exists a minimax composite polynomial $f_k \circ f_{k-1} \circ \cdots \circ f_2$ satisfying

$$\begin{aligned} &f_k \circ f_{k-1} \circ \cdots \circ f_2([1-\tau'', 1+\tau'']) \subseteq [1-t, 1+t] \\ &\mathsf{MultNum}(\{f_i\}_{2\le i\le k}) \le m - \mathsf{mult}(2i+1) \\ &\mathsf{DepNum}(\{f_i\}_{2\le i\le k}) \le n - \mathsf{dep}(2i+1). \end{aligned}$$

Let f_1 be the minimax approximate polynomial of degree at most $2i+1$ on $[1-\mathsf{InvMinErr}(2i+1,\tau''), 1+\mathsf{InvMinErr}(2i+1,\tau'')]$. Since $f_1([1-\mathsf{InvMinErr}(2i+1,\tau''), 1+\mathsf{InvMinErr}(2i+1,\tau'')]) = [1-\tau'', 1+\tau'']$, it holds that $f_k \circ f_{k-1} \circ \cdots \circ f_1([1-\mathsf{InvMinErr}(2i+1,\tau''), 1+\mathsf{InvMinErr}(2i+1,\tau'')]) \subseteq [1-t, 1+t]$. Also, it holds that $\mathsf{MultNum}(f_k \circ f_{k-1} \circ \cdots \circ f_1) \le m$ and $\mathsf{DepNum}(f_k \circ f_{k-1} \circ \cdots \circ f_1) \le n$. Thus, $\tau = f(m,n,t) < \mathsf{InvMinErr}(2i+1,\tau'')$, which is a contradiction. Thus, it holds that

$$\tau = \max_{\substack{1\le k \\ \mathsf{mult}(2k+1)\le m \\ \mathsf{dep}(2k+1)\le n}} \mathsf{InvMinErr}(2k+1, f(m-\mathsf{mult}(2k+1), n-\mathsf{dep}(2k+1), t)) \quad (1)$$

and the theorem is proved. □

$f(m,n,t)$ and $G(m,n,t)$ are recursively computed by the following Algorithm 3. In the 9th line of Algorithm 3, $\{2j+1\} \cup G(m-\mathsf{mult}(2j+1), n-\mathsf{dep}(2j+1), t)$ means inserting $2j+1$ to the ordered set $G(m-\mathsf{mult}(2j+1), n-\mathsf{dep}(2j+1), t)$ as the first component. In this paper, only minimax approximate polynomials of degree at most 31 are used to reduce the time complexity of the proposed algorithms. Since numerical results show that only minimax approximate polynomials of degree at most 11 are used to minimize the number of non-scalar multiplications, it seems that the minimax approximate polynomials of degree at most 31 are sufficient when minimizing the number of non-scalar multiplications. On the other hand, minimax approximate polynomials of large degree are sometimes used when minimizing the depth consumption. Thus, if minimax approximate polynomials of degree larger than 31 are also used, the required depth consumption may be further reduced.

Now, `DynMinMult` and `DynMinDep` algorithms are introduced, which use the values of $f(m,n,t)$ and $G(m,n,t)$ obtained from Algorithm 3. The following two cases are considered, which correspond to `DynMinMult` and `DynMinDep`, respectively.

First, `DynMinMult` puts more priority on minimizing the number of non-scalar multiplications rather than minimizing the depth consumption. The minimum number of non-scalar multiplications, M_{mult} is obtained. M_{dep} is the minimum required depth consumption among minimax composite polynomials that have the minimum number of non-scalar multiplications.

Algorithm 3: Computation of $f(m,n,t)$ and $G(m,n,t)$ using dynamic programming

Input: $t, m_{\max}, n_{\max}$
Output: $f(m,n,t), G(m,n,t)$ for $0 \le m \le m_{\max}$ and $0 \le n \le n_{\max}$
Generate 2-dimensional table $G(m,n,t)$ for $0 \le m \le m_{\max}$ and $0 \le n \le n_{\max}$, where the components are all empty sets.
for $m \leftarrow 0$ **to** $m_{\max}$ **do**
  for $n \leftarrow 0$ **to** $n_{\max}$ **do**
    if $m \le 1$ or $n \le 1$ **then**
      $f(m,n,t) \leftarrow t$
    else
      $j \leftarrow \underset{\substack{1 \le k \\ \mathsf{mult}(2k+1) \le m \\ \mathsf{dep}(2k+1) \le n}}{\operatorname{argmax}} \mathsf{InvMinErr}(2k+1, f(m-\mathsf{mult}(2k+1), n-\mathsf{dep}(2k+1), t))$
      $f(m,n,t) \leftarrow \mathsf{InvMinErr}(2j+1, f(m-\mathsf{mult}(2j+1), n-\mathsf{dep}(2j+1), t))$
      $G(m,n,t) \leftarrow \{2j+1\} \cup G(m-\mathsf{mult}(2j+1), n-\mathsf{dep}(2j+1), t)$
    end
  end
end

Second, `DynMinDep` puts more priority on minimizing the depth consumption rather than minimizing the number of non-scalar multiplications. The minimum depth consumption D_{dep} is obtained. D_{mult} is the minimum number of required non-scalar multiplications among minimax composite polynomials that have the minimum depth consumption.

$m_{\max}$ and $n_{\max}$ should be large enough to guarantee that the proposed algorithms find the minimax composite polynomial on R_δ that requires the minimum number of non-scalar multiplications and depth consumption among all $(\alpha-1,\delta)$-two-sided-close minimax composite polynomials on R_δ. $m_{\max}$ and $n_{\max}$ should satisfy $f(m_{\max}, n_{\max}, 2^{1-\alpha}) \ge \delta$ and we set $m_{\max}$ and $n_{\max}$ heuristically. Note that $\mathsf{dep}(d) \le \mathsf{mult}(d) \le 2\mathsf{dep}(d)$ for odd d less than or equal to 31. In [8], homomorphic comparison operations were proposed for cases when $\epsilon = 2^{-\alpha}$ and $\delta = \frac{1-\epsilon}{1+\epsilon}$ and we can use the minimum number of non-scalar multiplications (or depth consumption) values as in Table 2 to set $m_{\max}$ and $n_{\max}$ since we also propose homomorphic comparison operations for the same case in this paper. Let $q(\alpha)$ be the minimum number of non-scalar multiplications (or depth consumption) for the previous algorithms. We set $m_{\max} = n_{\max} = q(\alpha)$ when minimizing the number of non-scalar multiplications and set $m_{\max} = 2q(\alpha), n_{\max} = q(\alpha)$ when minimizing the depth consumption. Then, it holds that $f(m_{\max}, n_{\max}, 2^{1-\alpha}) \ge \delta$.

M_{degs} and D_{degs} are ordered sets that store the degrees of the component minimax approximate polynomials of corresponding optimal composite polynomial when minimizing the number of non-scalar multiplications and the depth consumption, respectively. Values of $M_{\mathrm{mult}}, M_{\mathrm{dep}}$, and M_{degs} can be obtained by

using Algorithm 4. Values of $D_{\text{mult}}, D_{\text{dep}}$, and D_{degs} can be obtained by using Algorithm 5. The procedure to find the optimal minimax composite polynomial using dynamic programming is summarized as follows:

(i) $f(m,n,t)$ and $G(m,n,t)$ are computed recursively using dynamic programming in Algorithm 3.
(ii) From the values of $f(m,n,t)$ and $G(m,n,t)$, find $M_{\text{mult}}, M_{\text{dep}}$, and M_{degs}, or D_{mult}, D_{dep}, and D_{degs} in Algorithms 4 and 5, respectively.
(iii) Find the component minimax approximate polynomials f_i's using modified Remez algorithm with M_{degs} or D_{degs}.

Algorithm 4: `DynMinMult`

Input: $\alpha, \delta, m_{\max}, n_{\max}, f(m,n,2^{1-\alpha}), G(m,n,2^{1-\alpha})$ for $0 \le m \le m_{\max}, 0 \le n \le n_{\max}$
Output: M_{mult}, M_{dep}, M_{degs}

```
1  for i ← 0 to m_max do
2      if f(i, n_max, 2^{1−α}) ≥ δ then
3          M_mult ← i
4          Go to line 7
5      end
6  end
7  for j ← 0 to n_max do
8      if f(M_mult, j, 2^{1−α}) ≥ δ then
9          M_dep ← j
10         Go to line 13
11     end
12 end
13 M_degs ← G(M_mult, M_dep, 2^{1−α})        // M_degs: ordered set
```

Theorem 4. *Let M_{mult}, M_{dep}, and M_{degs} be the output values of the* `DynMinMult` *algorithm in Algorithm 4 for inputs α and δ. Then, $M_{\text{mult}} \le \mathsf{MultNum}(\{f_i\}_{1\le i\le k})$ for any $(\alpha-1,\delta)$-two-sided-close composite polynomial of component polynomials with odd degree terms $f_k \circ f_{k-1} \circ \cdots \circ f_1$. In addition, if $M_{\text{mult}} = \mathsf{MultNum}(\{f_i\}_{1\le i\le k})$, then it holds that $M_{\text{dep}} \le \mathsf{DepNum}(\{f_i\}_{1\le i\le k})$.*

Proof. Let $f_k \circ f_{k-1} \circ \cdots \circ f_1$ be any $(\alpha-1,\delta)$-two-sided-close composite polynomial of component polynomials with odd degree terms. Let $\mathsf{MultNum}(\{f_i\}_{1\le i\le k}) = m$ and $\mathsf{DepNum}(\{f_i\}_{1\le i\le k}) = n$. From Theorem 2, there is a $(\alpha-1,\delta)$-two-sided-close minimax composite polynomial $\hat{f}_k \circ \hat{f}_{k-1} \circ \cdots \circ \hat{f}_1$ on R_δ such that $\mathsf{MultNum}(\{\hat{f}_i\}_{1\le i\le k}) \le \mathsf{MultNum}(\{f_i\}_{1\le i\le k})$ and $\mathsf{DepNum}(\{\hat{f}_i\}_{1\le i\le k}) \le \mathsf{DepNum}(\{f_i\}_{1\le i\le k})$. Assume that $m < M_{\text{mult}}$. Then, $\mathsf{MultNum}(\{\hat{f}_i\}) \le \mathsf{MultNum}(\{f_i\}) = m < M_{\text{mult}}$. Since $m < M_{\text{mult}}$ holds

Algorithm 5: `DynMinDep`

Input: $\alpha, \delta, m_{\max}, n_{\max}, f(m, n, 2^{1-\alpha}), G(m, n, 2^{1-\alpha})$ for $0 \le m \le m_{\max}, 0 \le n \le n_{\max}$
Output: D_{mult}, D_{dep}, D_{degs}

1 **for** $i \leftarrow 0$ **to** $n_{\max}$ **do**
2 **if** $f(m_{\max}, i, 2^{1-\alpha}) \ge \delta$ **then**
3 $D_{\mathrm{dep}} \leftarrow i$
4 Go to line 7
5 **end**
6 **end**
7 **for** $j \leftarrow 0$ **to** $m_{\max}$ **do**
8 **if** $f(j, D_{\mathrm{dep}}, 2^{1-\alpha}) \ge \delta$ **then**
9 $D_{\mathrm{mult}} \leftarrow j$
10 Go to line 13
11 **end**
12 **end**
13 $D_{\mathrm{degs}} \leftarrow G(D_{\mathrm{mult}}, D_{\mathrm{dep}}, 2^{1-\alpha})$ // D_{degs}: `ordered set`

and M_{mult} is the minimum i which satisfies $f(i, n_{\max}, 2^{1-\alpha}) \ge \delta$, it holds that $f(m, n_{\max}, 2^{1-\alpha}) < \delta$. Thus, there is no minimax composite polynomial $\bar{f}_k \circ \bar{f}_{k-1} \circ \cdots \circ \bar{f}_1$ on R_δ such that $\bar{f}_k \circ \bar{f}_{k-1} \circ \cdots \circ \bar{f}_1([1-\delta, 1+\delta]) \subseteq [1-2^{1-\alpha}, 1+2^{1-\alpha}]$, $\mathsf{MultNum}(\{\bar{f}_i\}_{1\le i\le k}) \le m$, and $\mathsf{DepNum}(\{\bar{f}_i\}_{1\le i\le k}) \le n_{\max}$. This leads to a contradiction since $\hat{\bar{f}}_k \circ \hat{\bar{f}}_{k-1} \circ \cdots \circ \hat{\bar{f}}_1$ is a $(\alpha-1, \delta)$-two-sided-close minimax composite polynomial on R_δ such that $\mathsf{MultNum}(\{\hat{f}_i\}_{1\le i\le k}) \le m$ and $\mathsf{DepNum}(\{\hat{f}_i\}_{1\le i\le k}) \le n_{\max}$.

In addition, assume that $M_{\mathrm{mult}} = m$ and $n < M_{\mathrm{dep}}$. Then, $f(m, n, 2^{1-\alpha}) < \delta$. Thus, there is no minimax composite polynomial $\bar{f}_k \circ \bar{f}_{k-1} \circ \cdots \circ \bar{f}_1$ on R_δ such that $\bar{f}_k \circ \bar{f}_{k-1} \circ \cdots \circ \bar{f}_1([1-\delta, 1+\delta]) \subseteq [1-2^{1-\alpha}, 1+2^{1-\alpha}]$, $\mathsf{MultNum}(\{\bar{f}_i\}_{1\le i\le k}) \le m$, and $\mathsf{DepNum}(\{\bar{f}_i\}_{1\le i\le k}) \le n$. This leads to a contradiction since $\hat{f}_k \circ \hat{f}_{k-1} \circ \cdots \circ \hat{f}_1$ is a $(\alpha-1, \delta)$-two-sided-close minimax composite polynomial on R_δ such that $\mathsf{MultNum}(\{\hat{f}_i\}_{1\le i\le k}) \le m$ and $\mathsf{DepNum}(\{\hat{f}_i\}_{1\le i\le k}) \le n$.

□

Theorem 5. *Let D_{mult}, D_{dep}, and D_{degs} be the output values of the* `DynMinDep` *algorithm in Algorithm 5 for inputs α and δ. Then, $D_{\mathrm{dep}} \le \mathsf{DepNum}(\{f_i\}_{1\le i\le k})$ any $(\alpha-1, \delta)$-two-sided-close composite polynomial of component polynomials with odd degree terms $f_k \circ f_{k-1} \circ \cdots \circ f_1$. In addition, if $D_{\mathrm{dep}} = \mathsf{DepNum}(\{f_i\}_{1\le i\le k})$, then it holds that $D_{\mathrm{mult}} \le \mathsf{MultNum}(\{f_i\}_{1\le i\le k})$.*

Proof. The proof is omitted because the proof of Theorem 5 is almost the same as that of Theroem 4. □

The `MinimaxComp` algorithm that outputs an approximate value of $\mathrm{comp}(a, b)$ is now proposed as in Algorithm 6, which uses the output M_{degs} of `DynMinMult` or the output D_{degs} of `DynMinDep` algorithm. $E(a, b; d)$ and $F(a, b; d)$ are defined for the description of the `MinimaxComp` algorithm as follows.

Definition 12. *For $a, b \in \mathbb{R}$ and $d \in \mathbb{N}$, let $F(a, b; d)$ be the minimax approximate polynomial of degree at most d on $[-b, -a] \cup [a, b]$ for $\mathrm{sgn}(x)$ and $E(a, b; d)$ be the minimax approximation error of the minimax approximate polynomial $F(a, b; d)$.*

Algorithm 6: `MinimaxComp`

Input: $a, b \in (0, 1)$, α, ϵ
Output: An approximate value of $\mathrm{comp}(a, b)$

$\{d_1, d_2, \cdots, d_k\} \leftarrow M_{\mathrm{degs}}$ from `DynMinMult` or D_{degs} from `DynMinDep` for α and $\delta = \frac{1-\epsilon}{1+\epsilon}$
$f_1 \leftarrow F(1-\epsilon, 1; d_1)$
$\tau_1 \leftarrow E(1-\epsilon, 1; d_1)$
for $i \leftarrow 2$ **to** k **do**
    $f_i \leftarrow F(1-\tau_{i-1}, 1+\tau_{i-1}; d_i)$
    $\tau_i \leftarrow E(1-\tau_{i-1}, 1+\tau_{i-1}; d_i)$
end
return $\frac{f_k \circ f_{k-1} \circ \cdots \circ f_1(a-b)+1}{2}$

4 Numerical Results

In this section, the number of non-scalar multiplications and the depth consumption of the proposed algorithms for the approximate polynomial for the sign function are compared to those of the previous algorithm [8].

4.1 Computation of the Required Non-Scalar Multiplications and Depth Consumption

Let s_f and s_g be the numbers of compositions of f_n and g_n, respectively. `NewCompG` algorithm in [8] approximates $\mathrm{comp}(a, b)$ using the composite polynomial $f_n^{(s_f)} \circ g_n^{(s_g)}$ with $n = 4$. According to Lemmas 1 and 3 in [8], if $s_g \geq \lceil \frac{1}{\log 0.98 c_n^2} \cdot \log(2/\epsilon) \rceil$ and $s_f \geq \lceil \frac{1}{\log(n+1)} \cdot \log(\alpha - 2) \rceil$, then the approximation error of the output of $\texttt{NewCompG}(a, b; n, s_f, s_g)$ compared to the value of $\mathrm{comp}(a, b)$ is upper bounded by $2^{-\alpha}$, where $c_n = \frac{2n+1}{4^n}\binom{2n}{n}$.

The previous approximation method for the sign function in [8] has the best performance for $n = 4$, where the degrees of the component approximate polynomials f_n and g_n are 9, and both the required numbers of non-scalar multiplications and the depth consumption for each component polynomial are 4. Then, it should hold that $s_g \geq \lceil 0.3894 \log(2/\epsilon) \rceil$ and $s_f \geq \lceil 0.4307 \log(\alpha - 2) \rceil$.

In this paper, the performances of the previous and proposed algorithms are analyzed when $\epsilon = 2^{-\alpha}$, which means that the input and output of the comparison operation are required to have the same precision bits. Then both the total required numbers of non-scalar multiplications and the depth consumption are at least $4(\lceil 0.3894(\alpha + 1) \rceil + \lceil 0.4307 \log(\alpha - 2) \rceil)$.

In the proposed method, the minimum number of required non-scalar multiplications and depth consumption are computed by using the `DynMinMult` and `DynMinDep` algorithms.

4.2 Comparisons

Table 2. Comparison of the minimum number of non-scalar multiplications and the corresponding depth consumption between the previous and the proposed algorithms while minimizing the number of non-scalar multiplications.

α	number of non-scalar multiplications		depth consumption	
	the previous algorithm	the proposed algorithm	the previous algorithm	the proposed algorithm
5	16	8	16	8
6	16	11	16	10
7	24	12	24	12
8	24	14	24	14
9	24	16	24	15
10	28	18	28	16
11	28	19	28	19
12	32	20	32	20
13	32	22	32	22
14	32	24	32	23
15	36	25	36	25
16	36	27	36	26
17	40	28	40	28
18	40	30	40	29
19	40	31	40	31
20	44	33	44	32

Table 3. Comparison of the minimum depth consumption and the corresponding number of non-scalar multiplications between the previous and the proposed algorithms while minimizing the depth consumption.

α	number of non-scalar multiplications		depth consumption	
	the previous algorithm	the proposed algorithm	the previous algorithm	the proposed algorithm
5	16	10	16	7
6	16	14	16	8
7	24	14	24	10
8	24	18	24	11
9	24	18	24	13
10	28	21	28	14
11	28	25	28	15
12	32	28	32	16
13	32	31	32	17
14	32	31	32	19
15	36	34	36	20
16	36	37	36	21
17	40	40	40	22
18	40	43	40	23
19	40	47	40	24
20	44	50	44	25

Table 4. The ordered sets M_{degs} and D_{degs} that store the degrees of the optimal component minimax approximate polynomials in `DynMinMult` and `DynMinDep` algorithms, respectively.

α	M_{degs}	D_{degs}
5	$\{9, 9\}$	$\{7, 13\}$
6	$\{5, 7, 9\}$	$\{15, 15\}$
7	$\{9, 9, 9\}$	$\{7, 7, 13\}$
8	$\{3, 9, 9, 9\}$	$\{7, 15, 15\}$
9	$\{7, 9, 9, 9\}$	$\{7, 7, 7, 13\}$
10	$\{3, 7, 7, 9, 9\}$	$\{7, 7, 13, 15\}$
11	$\{5, 9, 9, 9, 9\}$	$\{7, 7, 15, 31\}$
12	$\{9, 9, 9, 9, 9\}$	$\{7, 15, 15, 31\}$
13	$\{3, 9, 9, 9, 9, 9\}$	$\{15, 15, 15, 31\}$
14	$\{7, 9, 9, 9, 9, 9\}$	$\{7, 7, 13, 15, 31\}$
15	$\{3, 5, 9, 9, 9, 9, 9\}$	$\{13, 7, 15, 15, 31\}$
16	$\{5, 7, 9, 9, 9, 9, 9\}$	$\{13, 15, 15, 15, 31\}$
17	$\{9, 9, 9, 9, 9, 9, 9\}$	$\{13, 15, 15, 31, 31\}$
18	$\{7, 3, 9, 9, 9, 9, 9, 9\}$	$\{13, 15, 31, 31, 31\}$
19	$\{5, 9, 9, 9, 9, 9, 9, 9\}$	$\{15, 31, 31, 31, 31\}$
20	$\{9, 9, 9, 9, 9, 9, 9, 11\}$	$\{31, 31, 31, 31, 31\}$

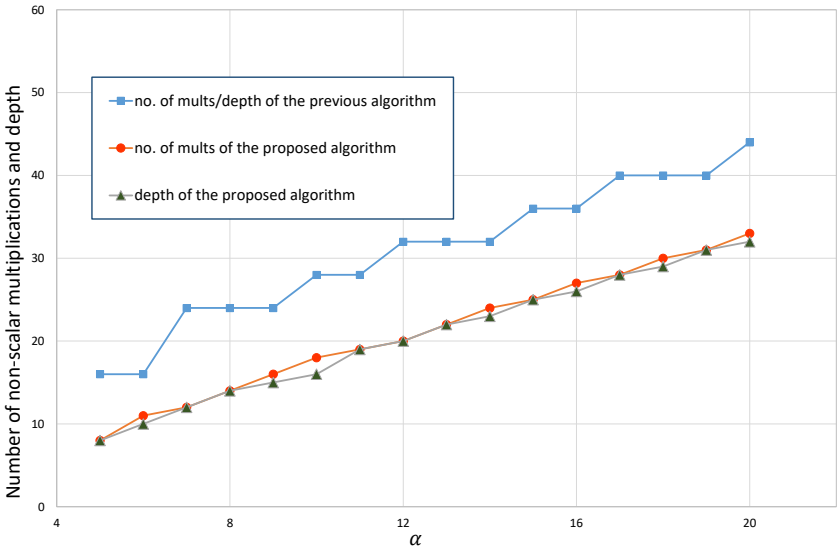

Fig. 2. Comparison of the minimum number of non-scalar multiplications and the corresponding depth consumption between the previous and the proposed algorithms while minimizing the number of non-scalar multiplications.

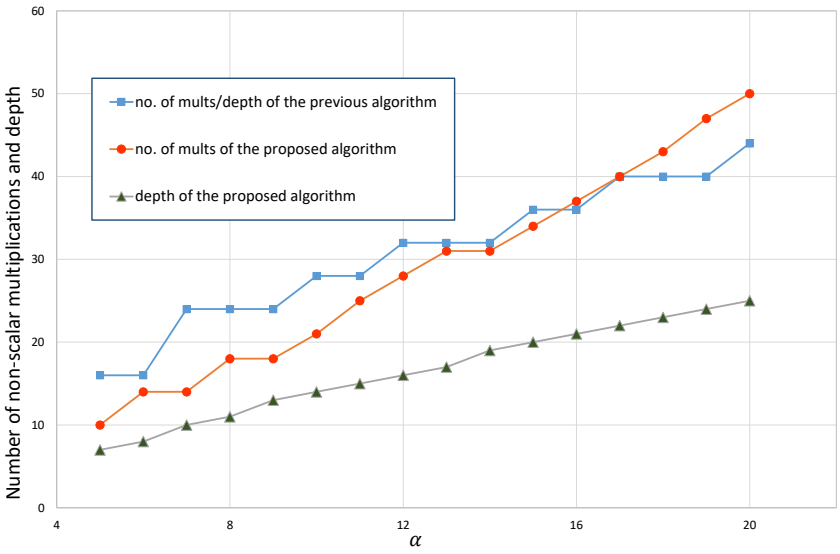

Fig. 3. Comparison of the minimum depth consumption and the corresponding number of non-scalar multiplications between the previous and the proposed algorithms while minimizing the depth consumption.

Table 2 shows the comparison of the minimum number of non-scalar multiplications and the corresponding depth consumption between the previous algorithm and the proposed algorithm `DynMinMult` while minimizing the number of non-scalar multiplications. It can be seen from Table 2 that the minimum number of the required non-scalar multiplications and the corresponding depth consumption for the proposed algorithms are reduced by about 33% and 35% on average, respectively, compared to those of the previous algorithm. The proposed algorithm `DynMinMult` intends to minimize the number of non-scalar multiplications, however, the corresponding depth consumption is also decreased. Figure 2 describes Table 2 as a graph.

Table 3 shows the comparison of the minimum depth consumption and the corresponding number of non-scalar multiplications between the previous algorithm and the proposed algorithm `DynMinDep` while minimizing the depth consumption. It can be seen from Table 3 that the non-scalar multiplications and the corresponding depth consumption for the proposed algorithms are reduced by about 10% and 47% on average, respectively, compared to those of the previous algorithm. If $\alpha \geq 16$, then the number of non-scalar multiplications for the proposed algorithm is slightly larger than that for the previous algorithm. However, when bootstrapping is used, the proposed algorithm requires lower time complexity than the previous algorithm since bootstrapping due to large depth consumption requires higher time complexity than non-scalar multiplication operations. Figure 3 describes Table 3 as a graph.

Table 4 shows the ordered sets M_{degs} and D_{degs} that store the degrees of the optimal component minimax approximate polynomials when minimizing the number of non-scalar multiplications and depth consumption, respectively.

5 Conclusion

We proposed a new approximation method for the homomorphic comparison operation using minimax composite polynomials obtained by the modified Remez algorithm. Our main idea is to find the minimax composite polynomial on R_δ that requires the minimum number of non-scalar multiplications and depth consumption among all $(\alpha - 1, \delta)$-two-sided-close minimax composite polynomials on R_δ. It was proved that the obtained minimax composite polynomial on R_δ requires less number of non-scalar multiplications and depth consumption than any $(\alpha - 1, \delta)$-two-sided-close composite polynomial of component polynomials with odd degree terms. Since the brute-force search requires considerable time for α, we proposed polynomial-time algorithms that obtain the best minimax composite polynomials by using dynamic programming. It can be seen from numerical analysis that when the number of non-scalar multiplications is minimized, the minimum number of required non-scalar multiplications and the corresponding depth consumption for the proposed algorithm `DynMinMult` are reduced by about 33% and 35% on average, respectively, compared to those for the previous algorithm. In addition, when the depth consumption is minimized, the minimum number of required non-scalar multiplications and the correspond-

ing depth consumption for the proposed algorithm `DynMinDep` are reduced by about 10% and 47% on average, respectively, compared to those for the previous algorithm.

References

1. P. Martins, L. Sousa, and A. Mariano, "A survey on fully homomorphic encryption: an engineering perspective," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, pp. 1–33, 2017.
2. C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. of the Forty-First Annual ACM Symposium on Theory of Computing*, 2019, pp. 169–178.
3. Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) Fully homomorphic encryption without bootstrapping," *ACM Transactions on Computation Theory (TOCT)*, vol. 6, no. 3, pp. 1–36, 2014.
4. J. Fan and F. Vercautern, "Somewhat practical fully homomorphic encryption," Cryptol. ePrint Arch., Tech. Rep. 2012/144, 2012. [Online]. Available: https://eprint.iacr.org/2012/144.
5. I. Chillotti, N. Gama, M. Georgieva, and M. Izabachene, "TFHE: fast fully homomorphic encryption over the torus," *Journal of Cryptology*, vol. 33, no. 1, pp. 34–91, 2020.
6. J. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Proc. International Conference on the Theory and Application of Cryptology and Information Security*, LNCS, Berlin, Germany: Springer, 2017, pp. 409–437.
7. J. Cheon, D. Kim, D. Kim, H. Lee, and K. Lee, "Numerical method for comparison on homomorphically encrypted numbers," in *Proc. International Conference on the Theory and Application of Cryptology and Information Security*, LNCS, Berlin, Germany: Springer, 2019, pp. 415-445.
8. J. Cheon, D. Kim, and D. Kim, "Efficient homomorphic comparison methods with optimal complexity," Cryptol. ePrint Arch., Tech. Rep. 2019/1234, 2019. [Online]. Available: https://eprint.iacr.org/2019/1234.
9. D. Gorenstein, W. W. Peterson, and N. Zierler, "Two-error correcting Bose-Chaudhuri codes are quasi-perfect," *Information and Control*, vol. 3, no. 3, pp. 291–294, 1960.
10. E. Y. Remez, "Sur la determination des polynomes d'approximation de degre donnee," *Comm. Soc. Math. Kharkov*, vol. 10, no. 196, pp. 41–63, 1934.
11. J. Lee, E. Lee, Y. Lee, Y. Kim, and J. No, "Optimal minimax polynomial approximation of modular reduction for bootstrapping of approximate homomorphic encryption," Cryptol. ePrint Arch., Tech. Rep. 2020/552, 2020. [Online]. Available: https://eprint.iacr.org/2020/552.
12. E. W. Cheney, *Introduction to Approximation Theory*. Cambridge, U.K.:McGraw-Hill, 1966.
13. K. Han and D. Ki, "Better bootstrapping for approximate homomorphic encryption," in *Proc. Cryptographers' Track at the RSA Conference*, LNCS, Berlin, Germany: Springer, 2020, pp. 364–390.
14. M. S. Paterson and L. J. Stockmeyer, "On the number of nonscalar multiplications necessary to evaluate polynomials," *SIAM Journal on Computing*, vol. 2, pp. 60–66, 1973.

15. R. E. Goldschmidt, *Applications of Division by Convergence*. PhD thesis, Massachusetts Institute of Technology, 1964.
16. Y. Lee, J. Lee, Y. Kim, and J. No, "Near-optimal polynomial for modulus reduction using l2-norm for approximate homomorphic encryption," Cryptol. ePrint Arch., Tech. Rep. 2020/488, 2020. [Online]. Available: https://eprint.iacr.org/2020/488.
17. C. Boura, N. Gama, and M. Georgieva, "Chimera: a unified framework for B/FV, TFHE and HEAAN fully homomorphic encryption and predictions for deep learning," Cryptol. ePrint Arch., Tech. Rep. 2018/758, 2018. [Online]. Available: https://eprint.iacr.org/2018/758.
18. D. Chialva and A. Dooms, "Conditionals in homomorphic encryption and machine learning applications," Cryptol. ePrint Arch., Tech. Rep. 2018/1032, 2018. [Online]. Available: https://eprint.iacr.org/2018/1032.
19. D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. on Pattern Analysis & Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
20. J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.
21. R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in *Proc. International Conference on Machine Learning*, 2016, pp. 201–210.
22. C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, "GAZELLE: A low latency framework for secure neural network inference," in *Proc. 27th USENIX Secur. Symp. (USENIX Security)*, 2018, pp. 1651–1669.