IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 May 2025, accepted 26 August 2025, date of publication 4 September 2025, date of current version 12 September 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3606013

METHODS

Polynomial Function Approximations With Leading Integer Coefficients for Efficient Encrypted Implementations

DIETER TEICHRIB, (Member, IEEE), JANIS ADAMEK, (Graduate Student Member, IEEE), PHILIPP BINFET, AND MORITZ SCHULZE DARUP, (Senior Member, IEEE)
Control and Cyberphysical Systems Group, TU Dortmund University, 44227 Dortmund, Germany

Corresponding author: Dieter Teichrib (dieter.teichrib@tu-dortmund.de)

This work was supported by German Research Foundation (DFG) under Grant 422262716 and Grant 503491151.

ABSTRACT Computations on encrypted data can, in principle, be performed using homomorphic encryption. However, due to certain limitations, only algorithms based on polynomial functions can be efficiently implemented in an encrypted setting. Consequently, polynomial approximations of non-polynomial functions are essential for efficient encrypted computations. In particular, low- to moderate-degree polynomial approximations of activation functions in neural networks are of special interest. We show that the accuracy of *encryption-friendly* approximations can be improved through a simple yet effective extension of state-of-the-art methods. Specifically, we show that enforcing a leading integer coefficient enables the use of polynomials of one degree higher than all existing approaches. Incorporating this novel integer constraint into classical regression problems initially leads to mixed-integer programs (MIPs). However, we develop tailored solution schemes that avoid MIP solving. Using these schemes, we compute new polynomial approximations for various test cases and demonstrate the effectiveness of our method compared to existing approaches.

INDEX TERMS Polynomial regression, optimization, homomorphic encryption, Chebyshev regression, privacy-preserved machine learning.

I. INTRODUCTION AND PROBLEM STATEMENT

Homomorphic encryption (HE) enables computations on encrypted data (see [1] for an overview). The unique capability of encrypted computations has unlocked a wide range of fascinating real-world applications across various fields such as privacy-preserving machine learning (ML) [2], [3], secure cloud computing [4], encrypted database queries [5], [6], encrypted financial services [7], [8], secure energy grid management [9], [10], secure voting systems [11], and encrypted control of networked systems [12]. In all these applications, HE enables new exciting features such as learning on encrypted data, encrypted regression, encrypted classification, or encrypted decision making.

The associate editor coordinating the review of this manuscript and approving it for publication was Asadullah Shaikh.

Yet, while the number and performance of homomorphically encrypted applications is increasing, encrypted computations are still challenging. In fact, the set of available operations offered by HE schemes is typically quite limited and mainly includes encrypted multiplications and additions. Moreover, the number of consecutive encrypted multiplications (or, more precisely, the multiplicative depth) is usually restrictive [13]. These limitations hinder the encrypted evaluation of many functions or algorithms. Nevertheless, polynomials of moderate degree can be evaluated efficiently on encrypted data. As a consequence, accurate polynomial approximations of non-polynomial functions are currently intensively investigated in the context of encrypted implementations [14], [15]. In particular, due to their heavy usage in ML, popular activation functions in artificial neural networks such as rectified linear units (ReLU) [3], [16],

sigmoid functions [17], or hyperbolic tangent are of special interest. The recent ReLU approximation challenge [18] underlines the significance of this research field even for commercial applications. In addition, polynomial approximations are relevant for further methods in encrypted ML, like logistic regression [19], [20] and are also utilized in other fields such as encrypted control [21].

Given the demand for effective polynomial approximations in the context of HE, our problem of interest can be formalized as follows. We aim for *encryption-friendly* polynomials of the form

$$p(x) := c_0 + c_1 x + \cdots + c_n x^n \tag{1}$$

approximating functions $f(x)$ on closed, non-empty intervals $[a, b]$. Now, for implementing (1) in an encrypted fashion using an HE scheme allowing for a multiplicative depth $d \in \mathbb{N}$, a tree-based realization – also known as exponentiation by squaring – is the current standard. It allows for implementing monomials of degree up to $n = 2^d - 1$ with generic coefficients $c_i \in \mathbb{R}$ [22], as illustrated in Figure 1.a for the example of c_3x^3 and $d = 2$. In this paper, we propose a new method for increasing the degree of polynomials that can be implemented in an encrypted fashion under a multiplicative depth d from $2^d - 1$ to 2^d. For moderate multiplicative depths such as $d \in \{1, 2, \ldots, 5\}$, this additional degree can make a significant difference as we illustrate with various numerical examples in Section IV. To the best of the authors' knowledge, such an approach has not been considered in the existing literature.

Technically, our method builds on restricting the polynomial's leading coefficient c_n to the set of integers $\mathbb{Z}$. In fact, as illustrated in Figure 1.b and detailed in Section II-A, $c_n x^n$ can then be evaluated via $x^n + \cdots + x^n$ with no additional multiplication. Now, as specified in Section II-B, searching for optimal $c_n \in \mathbb{Z}$ in the context of regression problems initially leads to mixed-integer programs (MIPs). However, we show that the resulting MIPs can be solved efficiently by solving only continuous (i.e., non-integer) optimization problems (OP). Remarkably, as presented in Section III, our method can also be adapted to Chebyshev regression, which significantly extends its applicability. In summary, our main contributions are (i) a novel approach for encryption-friendly polynomial approximations building on leading integer coefficients, (ii) an efficient procedure for identifying optimal $c_n \in \mathbb{Z}$, and (iii) an illustration of the effectiveness of our method with various numerical examples. Before detailing our approach in the following sections, we briefly specify relevant notation.

Notation. We denote natural numbers, the set of integers, and real numbers by $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{R}$, respectively. Furthermore, with $\lfloor \cdot \rfloor$, $\lceil \cdot \rceil$, and $\lfloor \cdot \rceil$, we refer to rounding down, rounding up, and rounding to the nearest integer (with rounding towards zero in case of tie).

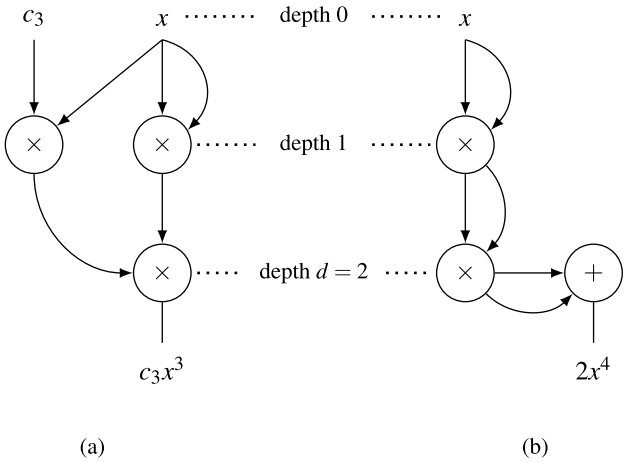

FIGURE 1. Computational circuits with multiplicative depth $d = 2$ for (a) a monomial of degree $3 = 2^d - 1$ with a generic coefficient and (b) a monomial of degree $4 = 2^d$ with an integer coefficient.

II. OPTIMAL POLYNOMIAL APPROXIMATIONS WITH LEADING INTEGER COEFFICIENTS

As summarized in the introduction, we are aiming for effective polynomial function approximations of the form (1) in the context of encrypted implementations. We briefly motivated that, given this special context, leading integer coefficients are beneficial. We specify this benefit in Section II-A before discussing the resulting regression problem and our proposed solution in Section II-B.

A. BENEFIT OF LEADING INTEGER COEFFICIENT

In HE schemes, the multiplicative depth $d \in \mathbb{N}$ is typically limited. It refers to the maximum number of consecutive encrypted multiplications supported by the given scheme. More precisely, assuming the desired computations are specified by a computational circuit as in Figure 1, then d is an upper limit for the number of multiplications (i.e., nodes "×" in Fig. 1) along any directed branch from an input node to an output node. If this limit is exceeded, computations usually lead to (highly) erroneous results. As a consequence, sticking to the limit is mandatory in encrypted computations.

Unsurprisingly, a limited multiplicative depth implies a limit on implementable monomials $c_n x^n$. For instance, naively computing $c_n x^n$ via $c_n \times x \times \cdots \times x$ requires n consecutive multiplications. The number of consecutive multiplications required to evaluate a monomial $c_n x^n$ can be easily reduced in optimized implementations. It is well-known that, for generic coefficients $c_n \in \mathbb{R}$, a tree-based realization requires the fewest possible number of consecutive multiplications and allows evaluating monomials of degree up to $n = 2^d - 1$ given a multiplicative depth d. For instance, given $d = 2$, we can evaluate c_3x^3 via $(c_3 \times x) \times (x \times x)$ as illustrated in Figure 1.a. Analogously, given $d = 3$ and defining $t := (x \times x)$, we can evaluate c_7x^7 via $[(c_7 \times x) \times t] \times [t \times t]$, where terms in round

brackets are evaluated at depth 1 and terms in square brackets at depth 2 (cf. Fig. 1). Up to now, this approach has built the basis for polynomial approximations in homomorphically encrypted computations. However, an improved realization of polynomials of degree up to $n = 2^d$ is possible when the leading coefficient c_n is an integer, as proposed by our approach and detailed below.

In fact, if $c_n \in \mathbb{Z}$, then one of the consecutive multiplications for computing $c_n x^n$ can be avoided and replaced by additions (which are not limited by the multiplicative depth). For instance, after having computed x^n using the classical tree-based approach, $c_n x^n$ can be evaluated via $x^n + \cdots + x^n$. Remarkably, tree-based approaches can also be utilized to compute this summation more efficiently. For example, $4x^2$ can be calculated via $s + s$ with $s := x^2 + x^2$ (being computed as in Fig. 1.b) using only two consecutive additions instead of three as for the naive approach. In general, for integer c_n, $c_n x^n$ can be evaluated using $\lceil \log_2(c_n) \rceil$ additions, which is important if larger $c_n \in \mathbb{Z}$ are considered. In summary, integer c_n are beneficial for encrypted implementations of polynomials as they allow to increase the implementable degree by one. However, as discussed next, efficiently designing polynomials with $c_n \in \mathbb{Z}$ for approximating non-polynomial functions is non-trivial.

B. EFFICIENT POLYNOMIAL REGRESSION WITH LEADING INTEGER COEFFICIENT

In order to leverage the increased polynomial degree in polynomial regression, we have to ensure $c_n \in \mathbb{Z}$. In principle, including this novel constraint is straightforward. In fact, assume N sampling points x_i are given and consider any standard regression problem of the form

$$\min_{c_0,\dots,c_n} J(c_0, \dots, c_n, x_1, \dots, x_N). \tag{2}$$

Then, adding the integer constraint $c_n \in \mathbb{Z}$ results in the desired restriction. For standard performance measures such as the mean squared error (MSE)

$$J(c_0, \dots, c_n, x_1, \dots, x_N) := \frac{1}{N} \sum_{i=1}^{N} (f(x_i) - p(x_i))^2, \tag{3}$$

the resulting optimization problem is a mixed-integer quadratic program (MIQP), which can be solved using standard software such as MOSEK [23] or Gurobi [24]. Yet, as we show next, the optimal solution can be derived more efficiently without relying on mixed-integer optimization. In fact, by exploiting that only one decision variable is restricted to integers, an optimal solution can be found by solving at most three continuous OP. An appropriate approach can be derived from the following theorem, whose proof is provided in the appendix.

Theorem 1: Assume the cost function J in (2) is convex in the coefficients c_i and assume $\boldsymbol{c}^* := (c_0^*, \dots, c_n^*)$ is an optimizer for (2) (without constraints). Then, there exists an optimizer $\hat{\boldsymbol{c}}^* := (\hat{c}_0^*, \dots, \hat{c}_n^*)$ for the constrained problem

$$\min_{\hat{c}_0,\dots,\hat{c}_n} J(\hat{c}_0, \dots, \hat{c}_n, x_1, \dots, x_N) \quad \text{s.t.} \quad \hat{c}_n \in \mathbb{Z}. \tag{4}$$

with $\hat{c}_n^* = \lfloor c_n^* \rfloor$ or $\hat{c}_n^* = \lceil c_n^* \rceil$.

From Theorem 1, we infer the following procedure for solving the MIP (4). Initially, we solve the unconstrained (and continuous) OP (2). If the resulting optimizer $\boldsymbol{c}^*$ is such that $c_n^* \in \mathbb{Z}$, we immediately found a solution to (4). Otherwise, we solve two variants of the OP (2), where we fix c_n to $\lfloor c_n^* \rfloor$ or $\lceil c_n^* \rceil$, respectively. The solution with the smaller cost function value then reflects a solution to (4). Hence, the MIP (4) can indeed be solved by (at most) three continuous OPs. As we show next, the procedure can even be shortened if J is symmetric with respect to the unconstrained optimizer $\boldsymbol{c}^*$. Among others, this is the case for the MSE (3) and the mean absolute error (MAE) considered further below in (13). A formal proof for the following theorem is (again) provided in the appendix.

Theorem 2: Let J and $\boldsymbol{c}^*$ be as in Theorem 1. Additionally, let J possess the point symmetry $J(\boldsymbol{c}^* + \boldsymbol{c}) = J(\boldsymbol{c}^* - \boldsymbol{c})$ for every $\boldsymbol{c} \in \mathbb{R}^{n+1}$. Then, there exists an optimizer $\hat{\boldsymbol{c}}^*$ for (4) with $\hat{c}_n^* = \lfloor c_n^* \rceil$.

Clearly, given the symmetry in Theorem 2, we can find a solution to (4) by solving at most two continuous OPs. In fact, assuming the solution to the unconstrained OP (2) is such that $c_n^* \notin \mathbb{Z}$, we only need to solve one additional variant of (2), where we fix $c_n := \lfloor c_n^* \rceil$. We conclude this section by formalizing that solving (4) for a certain degree n can only lead to an improvement (or tie) compared to the optimal solution of the unconstrained OP (2) for a smaller degree.

Corollary 1: Consider any positive $\hat{n} \in \mathbb{N}$ and let $\hat{\boldsymbol{c}}^*$ be an optimizer for (4) and $n := \hat{n}$. Furthermore, let $\boldsymbol{c}^*$ be an optimizer for (2) and an degree $n \in \mathbb{N}$ smaller than $\hat{n}$. Then, $J(\hat{\boldsymbol{c}}^*) \leq J(\boldsymbol{c}^*)$, where J refers to the same cost function instantiated for the two different degrees (but same number of sampling points N).

Proof: Let $\Delta n > 0$ be the difference of the degrees considered for the OPs (4) respectively (2). Then, the statement trivially follows from the fact that the optimizer $\boldsymbol{c}^*$ augmented by Δn zeros is feasible for (4) and $n := \hat{n}$. ∎

While Corollary 1 relates the two optimal cost function values through a non-strict inequality, in practice, we often observe that including the leading integer coefficient results in a significant improvement compared to unconstrained solutions of smaller degree. In such cases, we call the leading coefficient *beneficial* according to the following definition.

Definition 1: A leading integer coefficient is beneficial, if there exists an optimizer $\hat{\boldsymbol{c}}^*$ for (4) with $\hat{c}_n^* \neq 0$ but none with $\hat{c}_n^* = 0$.

We will illustrate and discuss the existence of beneficial leading integer coefficients with numerical examples in Section IV. Prior to this, we extend our results to Chebyshev regression in the following section.

III. EXTENSION TO CHEBYSHEV REGRESSION

From a numerical perspective, not only the integer constraint in (4) is challenging but also the relatively high polynomial degree n of (up to) 2^d. A standard approach addressing the latter issue is to consider more suitable polynomial families rather than the canonical power basis in (1). In particular, fitting Chebyshev polynomials of the form

$$\tilde{p}(z) := \tilde{c}_0 + \tilde{c}_1 T_1(z) + \cdots + \tilde{c}_n T_n(z) \tag{5}$$

with $T_i(z)$ referring to the i-th Chebyshev polynomial (of the first kind) typically works well for high polynomial degrees. This is mainly because Chebyshev regression avoids the large growth of higher-order coefficients as often observed in the canonical power basis. Still, using this approach requires to map the interval $[a, b]$ onto $[-1, 1]$ using, e.g., the mapping function

$$g(x) := \left(x - \frac{a+b}{2}\right)\frac{2}{b-a}. \tag{6}$$

Formally, (2) is then substituted by

$$\min_{\tilde{c}_0,\ldots,\tilde{c}_n} \tilde{J}\left(\tilde{c}_0, \ldots, \tilde{c}_n, g(x_1), \ldots, g(x_N)\right) \tag{7}$$

and, for MSE-based cost, (3) is replaced by

$$\tilde{J}(\tilde{c}_0, \ldots, \tilde{c}_n, z_1, \ldots, z_N) := \frac{1}{N}\sum_{i=1}^{N}\left(f(g^{-1}(z_i)) - \tilde{p}(z_i)\right)^2$$

with

$$g^{-1}(z) := \frac{b-a}{2}z + \frac{a+b}{2}$$

being the inverse mapping to (6). Now, solving (7) leads to an optimizer $\tilde{\boldsymbol{c}}^*$. For the implementation of the corresponding polynomial approximation of f, two scenarios are possible: First, given a sample x, one can compute $z := g(x)$ via (6) and, subsequently, evaluate (5). Second, one can expand $\tilde{p}(g(x))$ in x and, subsequently, evaluate the resulting polynomial for the given x.

For the encrypted implementation considered here, the first approach is unsuitable. In fact, mapping x via g would require a multiplication and, hence, would consume one level of the available multiplicative depth. Thus, we here focus on the second approach. Regarding the expansion, it is well known that only the polynomial $T_n(z)$ involves a monomial of the highest degree n. More specifically, this monomial reads $2^{n-1}z^n$. Taking the mapping $z = g(x)$ into account, we find that the leading coefficient of the expanded polynomial (of the form (1)) is given by

$$c_n = \tilde{c}_n 2^{n-1}\left(\frac{2}{b-a}\right)^n = \frac{\tilde{c}_n}{2}\left(\frac{4}{b-a}\right)^n. \tag{8}$$

Combining this observation with our integer constraint $c_n \in \mathbb{Z}$ from above, leads to the adapted constraint

$$\frac{\tilde{c}_n}{2}\left(\frac{4}{b-a}\right)^n \in \mathbb{Z} \tag{9}$$

for the OP (7). Clearly, this constraint is trivially fulfilled for $\tilde{c}_n = 0$. However, in this case, we also obtain $c_n = 0$, i.e., a polynomial of degree $n-1$. Satisfying the constraint (9) for $\tilde{c}_n \neq 0$ is slightly delicate. In fact, for moderate to high degrees n, the unconstrained OP (7) typically leads to $\tilde{c}_n^*$ with an absolute value (significantly) smaller than 1. This can be problematic due to relations between $\tilde{c}_n^*$ and $\hat{c}_n^*$ implied by Theorems 1 and 2. In fact, assuming Theorem 2 applies, a beneficial leading integer coefficient $\hat{c}_n^* \in \mathbb{Z} \setminus \{0\}$ can only exist if

$$\frac{\tilde{c}_n^*}{2}\left(\frac{4}{b-a}\right)^n \notin [-0.5, 0.5] \tag{10}$$

since, otherwise, applying $\lfloor \cdot \rceil$ to the left-hand side of (10), results in $\hat{c}_n^* = 0$. Theorem 1 allows to derive a similar condition[1] for the existence of an optimizer $\hat{\boldsymbol{c}}^*$ with $\hat{c}_n^* \in \mathbb{Z} \setminus \{0\}$, which even applies to asymmetric J. Now, for small $\tilde{c}_n^* \in (-1, 1)$, condition (10) can only be satisfied for large factors $4^n/(b-a)^n > 1$, which requires

$$b - a < 4. \tag{11}$$

Remarkably, it turns out that (11) is indeed necessary for a beneficial leading integer coefficient for all test cases in the following section.

IV. NUMERICAL CASE STUDIES

As previously discussed, Corollary 1 implies that our method cannot be outperformed by any existing approach, as they are all limited to polynomial approximations of at least one degree lower. Yet, there are cases (specified below) where our method performs equally well as an existing approach. Consequently, the relation in Corollary 1 is tight and admits no further improvement. Thus, we need numerical experiments to demonstrate the practical benefit of our method. To this end, we evaluate our method on various test cases involving different functions, domains, performance measures, and multiplicative depths. With regard to functions to be approximated, we consider

$$f_1(x) := \max\{x, 0\}, \tag{12a}$$
$$f_2(x) := \tanh(5x), \tag{12b}$$
$$f_3(x) := \frac{1}{1+e^{-10x}} \tag{12c}$$

inspired by activation functions commonly used in artificial neural networks. In fact, (12a) refers to a ReLU, while (12b) and (12c) reflect scaled[2] versions of the hyperbolic tangent and sigmoid functions, respectively. Regarding function domains, we consider different intervals $[a, b]$ with $a < 0 < b$ (due to the link to activation functions) and $b - a \leq 4$ due to (11). More specifically, we investigate the intervals $\mathcal{I}_1 := [-2, 2]$, $\mathcal{I}_2 := [-1, 1]$, $\mathcal{I}_3 := [-0.5, 0.5]$, $\mathcal{I}_4 := [-2, 1]$,

[1] According to Thm. 1, the left-hand side of (10) being not contained in $(-1, 1)$ is sufficient for the existence of an optimizer $\hat{\boldsymbol{c}}^*$ with $\hat{c}_n^* \in \mathbb{Z} \setminus \{0\}$.

[2] The scaling is required since the unscaled hyperbolic tangent and sigmoid functions are almost linear on the domains considered for the test cases, which renders the approximation task rather trivial.

TABLE 1. Performance improvement (in %) for $f_1(x)$ and MSE.

d	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{I}_4$	$\mathcal{I}_5$	$\mathcal{I}_6$	$\mathcal{I}_7$
1	–	–	92.63	–	–	–	–
2	–	–	75.39	–	–	–	–
3	–	47.03	51.39	–	–	48.09	48.09
4	–	30.83	30.83	–	–	18.36	18.36
5	–	17.03	17.03	–	–	–	–

TABLE 2. Performance improvement (in %) for $f_1(x)$ and MAE.

d	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{I}_4$	$\mathcal{I}_5$	$\mathcal{I}_6$	$\mathcal{I}_7$
1	–	–	73.68	–	–	1.09	1.09
2	–	–	52.53	–	–	–	–
3	–	34.60	34.67	–	–	33.16	33.16
4	–	20.21	20.21	–	–	14.89	14.89
5	–	11.15	11.15	2.58	2.58	2.67	2.67

$\mathcal{I}_5 := [-1, 2]$, $\mathcal{I}_6 := [-1, 0.5]$, and $\mathcal{I}_7 := [-0.5, 1]$, where we note that the first three are symmetric (i.e, $a = -b$) whereas the others are not. As performance measures, we consider the MSE (3) and the MAE

$$J(c_0, \ldots, c_n, x_1, \ldots, x_N) := \frac{1}{N} \sum_{i=1}^{N} |f(x_i) - p(x_i)| \quad (13)$$

or, more precisely, their counterparts for Chebyshev regression. Regarding the multiplicative depth, we take $d \in \{1, 2, 3, 4, 5\}$ and the corresponding maximum polynomial degrees $n := 2^d$ into account. Finally, regarding the number of sampling points, we consider $N := 10n$ points on a regular grid in each case. In summary, we consider $3 \times 7 \times 2 \times 5 = 210$ different test cases (reflecting the different combinations of functions, domains, performance measures, and multiplicative depths).

For each test case, we perform a Chebyshev regression and solve (7) subject to (i) the integer constraint (9) and (ii) the restriction $\tilde{c}_n = 0$. For the former, given the symmetry of the performance measures, we use Theorem 2 for an efficient solution. Regarding the latter, we note that it reflects the solution to the unconstrained OP of degree $n-1$. Having solved both OPs, we compare the two resulting performances (i.e., optimal function values) and compute the relative performance improvement resulting from the leading integer coefficient. The corresponding results are presented in Tables 1 to 6, where each individual table contains data for one function and one performance measure (but all depths d and intervals $\mathcal{I}_j$). In each table, when there is no improvement for a specific test case, we write "–" instead of "0.0" in order to highlight the cases, where significant improvements are achieved.

Now, the data in the tables offers numerous insights. First, the results confirm that existing methods never outperform our method, although ties do occur. Second, we do not observe an improvement for the interval $\mathcal{I}_1$ in any test case. This result is in line with our analysis in Section III since condition (11) is violated for $\mathcal{I}_1$ (due to $b - a = 4$)

TABLE 3. Performance improvement (in %) for $f_2(x)$ and MSE.

d	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{I}_4$	$\mathcal{I}_5$	$\mathcal{I}_6$	$\mathcal{I}_7$
1	–	–	–	–	–	59.08	59.08
2	–	–	–	0.41	0.41	81.86	81.86
3	–	–	–	–	–	28.25	28.25
4	–	–	–	27.63	27.63	70.72	70.72
5	–	–	–	57.84	57.84	56.27	56.27

TABLE 4. Performance improvement (in %) for $f_2(x)$ and MAE.

d	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{I}_4$	$\mathcal{I}_5$	$\mathcal{I}_6$	$\mathcal{I}_7$
1	–	–	–	–	–	45.84	45.84
2	–	–	–	2.58	2.58	58.30	58.30
3	–	–	–	–	–	3.41	3.41
4	–	–	–	1.48	1.48	39.91	39.91
5	–	–	0.07	35.69	35.69	42.48	42.48

TABLE 5. Performance improvement (in %) for $f_3(x)$ and MSE.

d	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{I}_4$	$\mathcal{I}_5$	$\mathcal{I}_6$	$\mathcal{I}_7$
1	–	–	–	–	–	55.00	55.00
2	–	–	–	–	–	49.98	49.98
3	–	–	–	–	–	5.01	5.01
4	–	–	–	–	–	0.07	0.07
5	–	–	–	52.65	52.65	–	–

TABLE 6. Performance improvement (in %) for $f_3(x)$ and MAE.

d	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{I}_4$	$\mathcal{I}_5$	$\mathcal{I}_6$	$\mathcal{I}_7$
1	–	–	–	–	–	45.84	45.84
2	–	–	–	–	–	56.48	56.48
3	–	–	–	–	–	3.41	3.41
4	–	–	–	–	–	39.91	39.91
5	–	–	0.34	35.69	35.69	42.48	42.48

and since we indeed find small $\tilde{c}_n^* \in (-1, 1)$ in every test case. For all other intervals $\mathcal{I}_j$, there exist multiple test cases, where the leading integer coefficient is beneficial. In general, improvements are more frequent for smaller intervals (i.e., smaller $b - a$) and for higher multiplicative depths d implying higher degrees n. Both are reasonable with regard to condition (10). Another trend is that, although improvements are more likely for larger values of d, the extent of these improvements tends to decrease as d increases. This makes sense since the positive effect of an additional degree diminishes with increasing degrees. To underline the significance of the improvements, we visualized the comparison between our solution and the current state of the art for the MSE regression task of the ReLU function in Figure 2 for $d = 3$ on $\mathcal{I}_3$. The improvement of 51.39% in Table 1 due to the higher polynomial degree is visually detectable.

Apart from the general trends discussed above, we can observe some more specific trends for the individual functions and intervals. First, note that the performance measures for the intervals $\mathcal{I}_4$ and $\mathcal{I}_5$ respectively the intervals $\mathcal{I}_6$ and $\mathcal{I}_7$ are identical within each row of every table.

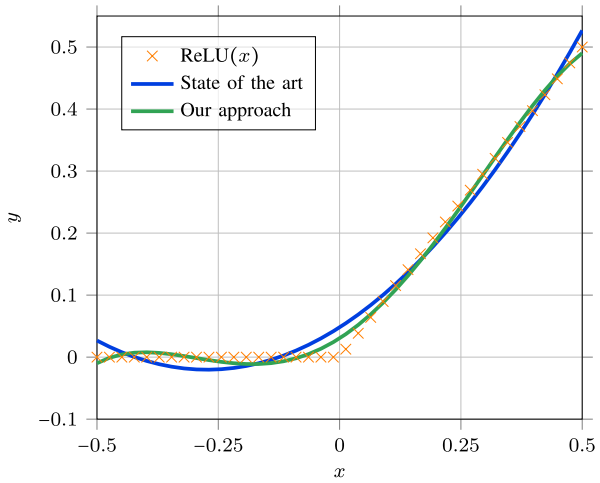

FIGURE 2. Exemplary comparison of our solution with the current state of the art for the MSE polynomial regression problem of the ReLU function with a multiplicative depth of $d = 3$ on $\mathcal{I}_3$.

This observation can be explained as follows. Clearly, the matching intervals can be transformed into each other by reflecting them across the origin. Now, the three test functions offer similar symmetries. In fact, it is easy to see that

$$f_1(x) = f_1(-x) + x, \tag{14a}$$
$$f_2(x) = -f_2(-x), \tag{14b}$$
$$f_3(x) = -f_3(-x) + 1 \tag{14c}$$

for every $x \in \mathbb{R}$. As a consequence, whenever we found a polynomial of degree $n \geq 1$ approximating one of the test functions f_i on a certain interval $[a, b]$ with a certain performance measure based on N data points on a regular grid, using the relations (14), we can derive a polynomial of the same degree which approximates f_i on $[-b, -a]$ with the same performance. In fact, for degrees $n \geq 1$, we can compensate for the "asymmetric offsets" x in (14a) respectively 1 in (14c). Hence, it is no surprise that the optimization-based results in all tables are equivalent for the matching intervals. The relations (14b) and (14c) further reveal that f_2 and $f_3(x) - 0.5$ are odd functions (whereas $f_1(x) - x/2 = |x|/2$ is even). Now, polynomial approximations of odd functions on symmetric intervals tend to yield odd polynomials. As a consequence, we here find $\tilde{c}_n^* \approx 0$ for almost every test case involving f_2 or f_3 and any of the intervals $\mathcal{I}_1$, $\mathcal{I}_2$, or $\mathcal{I}_3$. This explains the blocks of "–" in the Tables 3–6 (with only few exceptions for $d = 5$ potentially resulting from numerical limitations[3]). In summary, a beneficial leading integer coefficient was found in 86 of the 210 test cases. In 70 of the test cases (i.e., one third) the improvement exceeded 3%.

[3]Note that, for $d = 5$, we have $n = 2^d = 32$. Hence, computing c_n from $\tilde{c}_n$ according to (8) involves the factor $9.22 \cdot 10^{18}$. As a consequence, depending on the machine precision, we may find $\lfloor c_n^* \rceil \neq 0$ also for $\tilde{c}_n^* \approx 0$.

V. CONCLUSION AND OUTLOOK

We proposed a simple but effective method for more powerful polynomial function approximations in the context of HE. More precisely, we showed how to increase the supported polynomial degree (with respect to a limited multiplicative depth of the HE scheme) by one compared to state-of-the-art approaches. As illustrated with a comprehensive numerical case study involving popular activation functions for (deep) neural network-based machine learning, this additional degree can lead to significant improvements in approximation accuracy for low to moderate degree polynomials.

Technically, our approach builds on the consideration of polynomial approximations with leading integer coefficients c_n, where n is of the form 2^d with $d \in \mathbb{N}$. In fact, this feature enables the evaluation of monomials $c_n x^n$ in an encrypted fashion using a HE scheme supporting a multiplicative depth d (which was impossible before for more general $c_n \in \mathbb{R} \setminus \mathbb{Z}$). Performing a regression with the restriction $c_n \in \mathbb{Z}$ can be formulated as an MIP and, in principle, be solved using standard software. Yet, we showed that the solution to the MIP can also be obtained by solving at most two (for symmetric cost functions J) or three (for asymmetric J) continuous OP (see Thms. 1 and 2).

Future work aims for studying the beneficial effect of the novel polynomial approximations when used in compositions as it is the case, e.g., for privacy-preserving evaluations of deep neural networks or iterative optimization solvers.

APPENDIX
FORMAL PROOFS OF KEY RESULTS

Proof of Theorem 1: The statement of the theorem is trivially satisfied for the special case $c_n^* = \lfloor c_n^* \rceil = \lceil c_n^* \rceil \in \mathbb{Z}$. In all other cases, we have $\lfloor c_n^* \rceil < c_n^* < \lceil c_n^* \rceil$. For these cases, we consider a relaxed version of the OP (4), where the constraint $\hat{c}_n \in \mathbb{Z}$ is replaced by

$$\hat{c}_n \in \{c \in \mathbb{R} \mid c \leq \lfloor c_n^* \rceil\} \cup \{c \in \mathbb{R} \mid c \geq \lceil c_n^* \rceil\}. \tag{15}$$

Clearly, this OP can be solved by independently solving

$$\min_{\underline{c}_0, \ldots, \underline{c}_n} J(\underline{c}_0, \ldots, \underline{c}_n, x_1, \ldots, x_N) \quad \text{s.t.} \quad \underline{c}_n \leq \lfloor c_n^* \rceil \tag{16}$$

and

$$\min_{\overline{c}_0, \ldots, \overline{c}_n} J(\overline{c}_0, \ldots, \overline{c}_n, x_1, \ldots, x_N) \quad \text{s.t.} \quad \overline{c}_n \geq \lceil c_n^* \rceil, \tag{17}$$

and then selecting the solution with the smaller cost function value. Now, let $\underline{\boldsymbol{c}}^*$ and $\overline{\boldsymbol{c}}^*$ be optimizers for (16) and (17), respectively, and assume $J(\underline{\boldsymbol{c}}^*) \leq J(\overline{\boldsymbol{c}}^*)$, where $J(\boldsymbol{c})$ is short for $J(c_0, \ldots, c_n, x_1, \ldots, x_N)$. Then, existence of an optimizer $\hat{\boldsymbol{c}}^*$ for (4) satisfying $\hat{c}_n^* = \lfloor c_n^* \rceil$ can be shown as follows. First, convexity and feasibility of (16) implies that the Karush-Kuhn-Tucker (KKT) conditions are not only sufficient but also necessary for optimality. Hence, any

optimizer $\underline{\boldsymbol{c}}^*$ satisfies

$$\nabla_{\underline{\boldsymbol{c}}} J(\underline{\boldsymbol{c}}^*) + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \lambda^* \end{pmatrix} = \mathbf{0}, \tag{18a}$$

$$\underline{c}_n^* - \lfloor c_n^* \rfloor \leq 0, \tag{18b}$$

$$\lambda^* \geq 0, \tag{18c}$$

$$\lambda^*(\underline{c}_n^* - \lfloor c_n^* \rfloor) = 0 \tag{18d}$$

with $\lambda^* \in \mathbb{R}$ reflecting the (optimal) Lagrange multiplier. Next, taking (18c) into account, we distinguish the two cases (i) $\lambda^* > 0$ and (ii) $\lambda^* = 0$. In the first case, we immediately find $\underline{c}_n^* = \lfloor c_n^* \rfloor \in \mathbb{Z}$. Hence, $\underline{\boldsymbol{c}}^*$ is also feasible for (4). Moreover, since the set on the right-hand side in (15) is a superset of $\mathbb{Z}$, $J(\underline{\boldsymbol{c}}^*) \leq J(\overline{\boldsymbol{c}}^*)$ excludes the existence of an $\hat{\boldsymbol{c}}^*$ satisfying $J(\hat{\boldsymbol{c}}^*) < J(\underline{\boldsymbol{c}}^*)$. Hence, $\hat{\boldsymbol{c}}^* := \underline{\boldsymbol{c}}^*$ is an optimizer for (4) with the desired property (i.e., $\hat{c}_n^* = \lfloor c_n^* \rfloor$). It remains to consider the case $\lambda^* = 0$. In this case, (18a) implies $\nabla_{\underline{\boldsymbol{c}}} J(\underline{\boldsymbol{c}}^*) = \mathbf{0}$. Clearly, we also $\nabla_{\underline{\boldsymbol{c}}} J(\boldsymbol{c}^*) = \mathbf{0}$ for the unconstrained OP (2). Hence, due to convexity of J, we deduce $J(\underline{\boldsymbol{c}}^*) = J(\boldsymbol{c}^*)$. Furthermore, convexity implies

$$J((1-\alpha)\boldsymbol{c}^* + \alpha\underline{\boldsymbol{c}}^*) \leq (1-\alpha)J(\boldsymbol{c}^*) + \alpha J(\underline{\boldsymbol{c}}^*) = J(\boldsymbol{c}^*) \tag{19}$$

for every $\alpha \in [0, 1]$. Due to global optimality of $\boldsymbol{c}^*$, $J(\boldsymbol{c}^*)$ is also a lower bound for the left-hand side in (19) implying

$$J((1-\alpha)\boldsymbol{c}^* + \alpha\underline{\boldsymbol{c}}^*) = J(\boldsymbol{c}^*).$$

Now, taking $\underline{c}_n^* \leq \lfloor c_n^* \rfloor < c_n^*$ into account, it becomes clear that there exist an $\alpha \in (0, 1]$ such that

$$\hat{\boldsymbol{c}}^* := (1-\alpha)\boldsymbol{c}^* + \alpha\underline{\boldsymbol{c}}^*$$

satisfies $\hat{c}_n^* = \lfloor c_n^* \rfloor \in \mathbb{Z}$ and $J(\hat{\boldsymbol{c}}^*) = J(\boldsymbol{c}^*)$. Hence, $\hat{\boldsymbol{c}}^*$ is an optimizer for (4) with the desired property. Finally, the case $J(\underline{\boldsymbol{c}}^*) > J(\overline{\boldsymbol{c}}^*)$ leading to an optimizer for (4) with $\hat{c}_n^* = \lceil c_n^* \rceil$ can be handled analogously. ■

Proof of Theorem 2: We already know from Theorem 1 that there exists an optimizer $\hat{\boldsymbol{c}}^*$ for (4) with $\hat{c}_n^* = \lfloor c_n^* \rfloor$ or $\hat{c}_n^* = \lceil c_n^* \rceil$. Hence, it remains to show that, given the specified symmetry, out of the two options, the nearest integer $\lfloor c_n^* \rceil$ reflects a solution. To this end, we define the two (non-negative) quantities $\Delta\underline{c}_n := c_n^* - \lfloor c_n^* \rfloor$ and $\Delta\overline{c}_n := \lceil c_n^* \rceil - c_n^*$, and first consider $\Delta\underline{c}_n \leq \Delta\overline{c}_n$. Then, $\lfloor c_n^* \rceil = \lfloor c_n^* \rfloor$. In order to prove the claim, we next show that there does not exist a $\overline{\boldsymbol{c}}$ with $\overline{c}_n = \lceil c_n^* \rceil$ such that $J(\overline{\boldsymbol{c}}) < J(\underline{\boldsymbol{c}})$ for every $\underline{\boldsymbol{c}}$ with $\underline{c}_n = \lfloor c_n^* \rfloor$. To do so, we assume such a $\overline{\boldsymbol{c}}$ exists and derive a contradiction. Since $0 \leq \Delta\underline{c}_n \leq \Delta\overline{c}_n$, there exists an $\alpha \in [0, 1]$ such that $\alpha\Delta\overline{c}_n = \Delta\underline{c}_n$. Due to convexity of J, we further have

$$J((1-\alpha)\boldsymbol{c}^* + \alpha\overline{\boldsymbol{c}}) \leq (1-\alpha)J(\boldsymbol{c}^*) + \alpha J(\overline{\boldsymbol{c}}).$$

Taking $J(\boldsymbol{c}^*) \leq J(\overline{\boldsymbol{c}})$ due to optimality into account, we additionally obtain $J((1-\alpha)\boldsymbol{c}^* + \alpha\overline{\boldsymbol{c}}) \leq J(\overline{\boldsymbol{c}})$. Now, we define $\Delta\boldsymbol{c} := \overline{\boldsymbol{c}} - \boldsymbol{c}^*$ and note that $(1-\alpha)\boldsymbol{c}^* + \alpha\overline{\boldsymbol{c}} = \boldsymbol{c}^* + \alpha\Delta\boldsymbol{c}$. Thus, due to the symmetry of J, we find $J(\boldsymbol{c}^* + \alpha\Delta\boldsymbol{c}) = J(\boldsymbol{c}^* - \alpha\Delta\boldsymbol{c})$. It is easy to see that the $(n+1)$-th component of $\Delta\boldsymbol{c}$ equals $\Delta\overline{c}_n$. Thus, the $(n+1)$-th component of $\boldsymbol{c}^* - \alpha\Delta\boldsymbol{c}$ is $c_n^* - \alpha\Delta\overline{c}_n = c_n^* - \Delta\underline{c}_n = \lfloor c_n^* \rfloor$. In summary, $\underline{\boldsymbol{c}} := \boldsymbol{c}^* - \alpha\Delta\boldsymbol{c}$ is such that $\underline{c}_n = \lfloor c_n^* \rfloor$ and

$$J(\underline{\boldsymbol{c}}) = J(\boldsymbol{c}^* + \alpha\Delta\boldsymbol{c}) = J((1-\alpha)\boldsymbol{c}^* + \alpha\overline{\boldsymbol{c}}) \leq J(\overline{\boldsymbol{c}}),$$

which contradicts the assumption on $\overline{\boldsymbol{c}}$. The remaining case $\Delta\underline{c}_n > \Delta\overline{c}_n$ can be handled analogously. ■

ACKNOWLEDGMENT

(Dieter Teichrib and Janis Adamek contributed equally to this work.)

REFERENCES

[1] C. Marcolla, V. Sucasas, M. Manzano, R. Bassoli, F. H. P. Fitzek, and N. Aaraj, "Survey on fully homomorphic encryption, theory, and applications," *Proc. IEEE*, vol. 110, no. 10, pp. 1572–1609, Oct. 2022.

[2] R. Podschwadt, D. Takabi, P. Hu, M. H. Rafiei, and Z. Cai, "A survey of deep learning architectures for privacy-preserving machine learning with fully homomorphic encryption," *IEEE Access*, vol. 10, pp. 117477–117500, 2022.

[3] J.-W. Lee, H. Kang, Y. Lee, W. Choi, J. Eom, M. Deryabin, E. Lee, J. Lee, D. Yoo, Y.-S. Kim, and J.-S. No, "Privacy-preserving machine learning with fully homomorphic encryption for deep neural network," *IEEE Access*, vol. 10, pp. 30039–30054, 2022.

[4] S. J. Mohammed and D. B. Taha, "From cloud computing security towards homomorphic encryption: A comprehensive review," *TELKOMNIKA (Telecommun. Comput. Electron. Control)*, vol. 19, no. 4, pp. 1152–1161, Aug. 2021.

[5] W.-K. Lin, E. Mook, and D. Wichs, "Doubly efficient private information retrieval and fully homomorphic RAM computation from ring LWE," in *Proc. 55th Annu. ACM Symp. Theory Comput.*, Jun. 2023, pp. 595–608.

[6] D. Boneh, C. Gentry, S. Halevi, F. Wang, and D. J. Wu, "Private database queries using somewhat homomorphic encryption," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*, 2013, pp. 102–118.

[7] J. Byun, H. Ko, and J. Lee, "A privacy-preserving mean–variance optimal portfolio," *Finance Res. Lett.*, vol. 54, Jun. 2023, Art. no. 103794.

[8] V. S. Naresh and D. Ayyappa, "PPDNN-CRP: CKKS-FHE enabled privacy-preserving deep neural network processing for credit risk prediction," *Comput. Econ.*, pp. 1–25, Dec. 2024.

[9] Z.-P. Yuan, P. Li, Z.-L. Li, and J. Xia, "A fully distributed privacy-preserving energy management system for networked microgrid cluster based on homomorphic encryption," *IEEE Trans. Smart Grid*, vol. 15, no. 2, pp. 1735–1748, Mar. 2024.

[10] Z. Cheng, F. Ye, X. Cao, and M.-Y. Chow, "A homomorphic encryption-based private collaborative distributed energy management system," *IEEE Trans. Smart Grid*, vol. 12, no. 6, pp. 5233–5243, Nov. 2021.

[11] P. R. Naidu, D. R. Bolla, G. Prateek, S. S. Harshini, S. A. Hegde, and V. V. S. Harsha, "E-voting system using blockchain and homomorphic encryption," in *Proc. IEEE 2nd Mysore Sub Sect. Int. Conf. (MysuruCon)*, Oct. 2022, pp. 1–5.

[12] M. Schulze Darup, A. B. Alexandru, D. E. Quevedo, and G. J. Pappas, "Encrypted control for networked systems: An illustrative introduction and current challenges," *IEEE Control Syst. Mag.*, vol. 41, no. 3, pp. 58–78, Jun. 2021.

[13] N. Schlüter, P. Binfet, and M. Schulze Darup, "A brief survey on encrypted control: From the first to the second generation and beyond," *Annu. Rev. Control*, vol. 56, Jan. 2023, Art. no. 100913.

[14] E. Lee, J.-W. Lee, J.-S. No, and Y.-S. Kim, "Minimax approximation of sign function by composite polynomial for homomorphic comparison," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 6, pp. 3711–3727, Nov. 2022.

[15] S. Panda, "Polynomial approximation of inverse sqrt function for FHE," in *Proc. Int. Symp. Cyber Secur., Cryptol., Mach. Learn.*, 2022, pp. 366–376.

[16] J. Lee, E. Lee, J.-W. Lee, Y. Kim, Y.-S. Kim, and J.-S. No, "Precise approximation of convolutional neural networks for homomorphically encrypted data," *IEEE Access*, vol. 11, pp. 62062–62076, 2023.

[17] A. Kim, Y. Song, M. Kim, K. Lee, and J. H. Cheon, "Logistic regression model training based on the approximate homomorphic encryption," *BMC Med. Genomics*, vol. 11, no. S4, pp. 23–31, Oct. 2018.
[18] G. Arakelov, N. Kaskov, D. Pianykh, and Y. Polyakov, "FHERMA: Building the open-source FHE components library for practical use," *Cryptology ePrint Archive*, Apr. 2024. [Online]. Available: https://eprint.iacr.org/2024/612
[19] C. Bonte and F. Vercauteren, "Privacy-preserving logistic regression training," *BMC Med. Genomics*, vol. 11, no. S4, pp. 13–21, Oct. 2018.
[20] M. Tian, J. Liu, Z. Chen, and S. Wang, "Privacy-preserving logistic regression with improved efficiency," *J. Inf. Secur. Appl.*, vol. 85, Sep. 2024, Art. no. 103848.
[21] M. Kvasnica, J. Löfberg, and M. Fikar, "Stabilizing polynomial approximation of explicit MPC," *Automatica*, vol. 47, no. 10, pp. 2292–2297, Oct. 2011.
[22] K. Han and D. Ki, "Better bootstrapping for approximate homomorphic encryption," in *Proc. Cryptographers' Track RSA Conf.*, 2020, pp. 364–390.
[23] (2024). *The MOSEK Optimization Toolbox for MATLAB Manual*. [Online]. Available: http://docs.mosek.com/latest/toolbox/index.html
[24] Gurobi Optimization, LLC. (2024). *Gurobi Optimizer Reference Manual*. [Online]. Available: https://www.gurobi.com

DIETER TEICHRIB (Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from Paderborn University, in 2019 and 2020, respectively. He is currently pursuing the Ph.D. degree with the Control and Cyberphysical Systems Group, TU Dortmund University. His research interest includes tailoring machine learning methods for use in control.

JANIS ADAMEK (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from TU Dortmund University (TUD), in 2019 and 2022, respectively, where he is currently pursuing the Ph.D. degree with the Control and Cyberphysical Systems Group. His research interests include the development of privacy-enhanced optimization algorithms for control and machine learning.

PHILIPP BINFET received the B.Sc. and M.Sc. degrees in electrical engineering from Paderborn University, in 2019 and 2021, respectively. He is currently a Research Assistant with the Control and Cyberphysical Systems Group, TU Dortmund University. His research interests include secure computation and privacy-preserving optimization and control.

MORITZ SCHULZE DARUP (Senior Member, IEEE) received the Diploma degree in mechanical engineering, the B.Sc. degree in physics, and the Ph.D. degree in control engineering from Ruhr-Universitt Bochum, in 2008, 2010, and 2014, respectively. He was an Assistant Professor and the Head of an Emmy Noether Group at Paderborn University, in 2019, for encrypted control. Since 2020, he has been a Full Professor with the Control and Cyberphysical Systems Group, TU Dortmund University. His research interests include secure, predictive, and data-driven control.