RELU–FHE: LOW-COST ACCURATE RELU POLYNOMIAL APPROXIMATION IN FULLY HOMOMORPHIC ENCRYPTION BASED ML INFERENCE

Jingtian Dang* Jianming Tong* Anupam Golder Callie Hao Tushar Krishna
Georgia Institute of Technology

ABSTRACT

Machine learning (ML) is getting more pervasive. Wide adoption of ML in healthcare, facial recognition, and blockchain involves private and sensitive data. Inference on encrypted data, termed Fully Homomorphic Encryption (FHE), preserves the privacy of both data and the ML model. However, it slows down plaintext inference by six magnitudes, with a root cause of replacing non-linear operators with high-degree Polynomial Approximated Function (PAF). To reduce the degree without sacrificing accuracy, we propose (1) Coefficient Tuning (CT) to obtain a good initial coefficient value for each PAF, (2) Progressive Approximation (PA) to enable convergence by replacing ReLU and training parameters in a divide-and-conquer manner, and (3) Alternate Training (AT) to further improve the post-replacement accuracy. A combination of CT, PA, and AT enables the exploration of accuracy-latency space for FHE-domain ReLU replacement. Our evaluation shows that the optimal PAF with 12 degrees reduces 72% latency of the state-of-the-art 27-degree PAF with the same post-replacement accuracy (69.0%) on ResNet-18 using ImageNet 1k dataset.

1 INTRODUCTION

Machine learning (ML) is getting more pervasive with a wide deployment in healthcare (Mateen et al., 2020), facial recognition (Raji & Fried, 2021), and blockchain (Zhang et al., 2021), leading to privacy leakage with private and sensitive data involved. Fully Homomorphic Encryption (FHE) enables ML inference on encrypted data, preserving privacy from both data and models.

However, FHE-based ML inference comes with six magnitudes higher latency overheads than plaintext inference. Specifically, FHE-based ML inference requires the execution of both linear operators (Convolution, AveagePooling, Fully Connection, etc.) and non-linear operators (ReLU, MaxPooling). Surprisingly, around half of the latency is consumed by computation-intensive non-linear operators. The recurrent research problem is *how to efficiently process non-linear kernels in FHE.*

1.1 Challenges

The high latency of processing non-linear operators boils down to the fact that they are not naturally supported by FHE and demand other schemes or approximations.

A plethora of prior arts illustrate the infeasibility of resorting to other secure schemes for processing non-linear operators in the practical system because of prohibitive communication overheads for converting data securely among schemes (Gilad-Bachrach et al., 2016; Lou et al., 2021; Ran et al., 2022).

Additionally, approximations of non-linear kernels introduce the trade-off between accuracy and latency. An accurate approximation requires a high-degree Polynomial Approximated Function (PAF) with a prohibitive long chain of multiplication with bootstrapping. An example includes a SotA 27-degree PAF (Lee et al., 2021; Kim et al., 2022). On the other hand, a low-latency approximation suffers severe accuracy degradation, which requires a combination of mix-schemes and approximation to guarantee the accuracy (Lou et al., 2021). Both are suboptimal.

Further, prior arts also navigated the trade-off space between accuracy and latency by replacing ReLU with PAF followed by ML training-based coefficients fine-tuning. However, such a scheme hardly converge when for PAF with higher than 5 degrees, leading back to the inefficient mixing schemes paradigm with both other secure scheme and PAFs.

1.2 Our Contributions

To conquer the aforementioned limitations, this paper proposes the first-ever systematic method to replace all ReLU layers with low-degree PAFs without sacrificing accuracy (to the best of our knowledge, the comparison with prior arts is shown in Tab. 1).

Specifically, we propose three key techniques to decide the initial coefficients of PAF, progressively replace ReLU with PAFs and fine-tune coefficients of post-replacement PAFs as follows.

- In PAF coefficients decision, we propose *Coefficient*

Table 1. Comparison of proposed strategies with SotA.

	Low Communication Overhead	low accuracy degradation	low latency overhead
SafeNet, CryptoGCN	✗	✗	✓
CryptoNet, CryptoDL, LoLa, CHE	✗	✗	✓
F1, CraterLake, BTS	✓	✓	✗
HEAX,Delphi,Gazelle,Cheetah	✗	✗	✓
SHE	✓	✓	✗
ReLU-FHE	✓	✓	✓

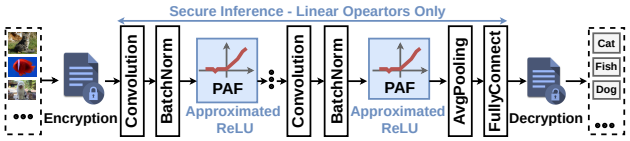

Figure 1. Overview of the FHE-base ML inference where original non-linear ReLU are approximated by linear Polynomial Approximated Activation (PAF).

Tuning (CT) that fine-tunes initial coefficients before ReLU replacement to improve post-replacement accuracy by $1.04\times \sim 2.38\times$ for PAFs of varying degrees.

- In ReLU replacement, we propose *Progressive Approximation* (PA) to optimize the whole deviation caused by polynomial approximation in a layer-wise divide-and-conquer manner, enabling the convergence of PAFs with an arbitrary degree and exploration of the full accuracy-latency tradeoff space.
- In post-replacement PAF coefficients fine-tunes, we propose *Alternate Training* (AT) to decouple and train weights and coefficients in an alternate fashion, resulting in repeat accuracy climbing by $1.009 \sim 1.037\times$ of the fine-tuned post-approximation ML model.
- A systematic scheduler to automatically explore accuracy-latency tradeoff space. Our results show that a sweet-point PAF with 12-degree can achieve the same 69.0% post-replacement accuracy with 72% latency reduction in ResNet-18 inference, compared to the SotA 27-degree PAF (Lee et al., 2021).

2 Technical Background

2.1 Non-linear Kernel in FHE-based ML Inference

Fully Homomorphic Encryption (FHE) is an asymmetric encryption scheme that enables ciphertext-based computation with CKKS as the mostly used FHE scheme for machine learning inference due to its superior efficiency in approximate computation compared to other schemes such as BGV, BFV, and TFHE (Riazi et al., 2020). Under the CKKS scheme, only linear operators are allowed as shown in Fig. 1 such that ReLU and MaxPooling need to be replaced by PAF.

2.2 Polynomial Approximated Function (PAF)

High-degree polynomials could approximate arbitrary functions theoretically. In practice, however, a direct approximation of ReLU using PAF introduces severe approximation errors (Lee et al., 2022). Instead, prior arts use polynomials to approximate the $sign(x)$ function, which outputs 1 if x is positive, -1 if x is negative, and 0 for zero. Then both ReLU and Max functions could be constructed respectively by $\frac{(x+sign(x)\cdot x)}{2}$ and $\frac{(x+y)+(x-y)\cdot sign(x-y)}{2}$.

Table 2. Baseline PAF and post-replacement validation accuracy

Form	$\alpha = 10$	$\alpha = 7$	$f_1^2 \circ g_1^2$	$f_2 \circ g_3$	$f_2 \circ g_2$	$f_1 \circ g_2$
Degree	27	14	12	12	10	8
Multiplication Depth	29	14	12	9	8	7
Accuracy	69.0	64.7	51.3	49.4	32	18.6

Lower latency and less accuracy degradation are two fundamental goals of replacing non-linear ReLU functions with Polynomial Approximated Functions (PAF). Latency is reflected in the multiplication depth of the polynomial, or the degree of the polynomial, as multiplication using fully homomorphic encryption (FHE) incurs a latency that is 3 orders of magnitude higher than FHE-based addition. While accuracy degradation arises from the deviation between the PAF and the original non-linear ReLU. This deviation is caused by the approximation error introduced by the PAF, which can vary depending on the degree of the PAF and the coefficients of the PAF.

To approximate the $sign(x)$ function, a cascaded polynomial achieves fewer errors compared to the single polynomial with the same multiplication depth (Lee et al., 2022; 2021), and we thus pick cascaded polynomial as baseline PAF with post-replacement accuracy shown in Tab. 2. A typical cascaded polynomial is represented by f^n, indicating a serial cascaded connection of n polynomial f.

3 Proposed Method

3.1 Overview

In this section, we propose three techniques to enable low latency (low degree) PAF with negligible post-replacement accuracy degradation. (a) Coefficient Tuning (CT) tunes each PAF to better fit with the original context of ReLU. (b) Progressive Approximation (PA) enables progressive deviation correction through neural network re-training instead of directly fixing the entire deviation by replacing all ReLUs with PAF once. (c) Alternate Training (AT) decouples the training of PAF coefficients and parameters of other layers for further accuracy improvement.

3.2 Coefficients Tuning (CT)

The naive approach for replacing non-linear rectified linear unit (ReLU) functions with polynomial approximated functions (PAF) is to directly replace all ReLU with a unified PAF. However, this approach does not take into account the fact that the distributions for input activations at different ReLU layers are different.

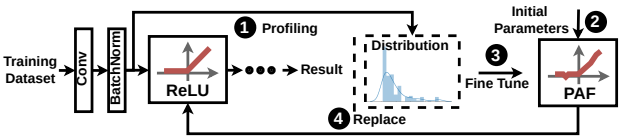

Figure 2. Coefficients Tuning (CT) uses profiled distribution to fine-tune PAF coefficients to generate more accurate results on a small value range with high probability in distribution.

Therefore, we propose Coefficients Tuning (CT) as a technique to use profiled data distribution to further fine-tune coefficients of low-degree PAF, and then perform the replacement with post-CT PAF, as shown in Fig. 2 Specifically, we sample data from the real distribution of profiled input activations of each individual ReLU layer, together with actual ReLU results as the labels. The sampled data is used to train initial PAFs and then post-CT PAFs are used to replace the corresponding ReLU.

Intuitively, a PAF with a lower degree may not replicate the ReLU output over the entire input range, but it can replicate the ReLU output more accurately on a reduced input range. CT renders low-degree polynomials only generate accurate results over a small input range that is the highest-probability range in the data distribution, and thus achieve better accuracy improvement. Besides, CT reduces training time because post-CT PAF's output is closer to the output of ReLU at the replacement point than initial PAFs.

3.3 Progressive Approximation (PA)

Direct replacement of all ReLU with PAFs during training may not recover degraded accuracy and may even introduce further accuracy degradation due to non-convergence. Because the approximation error between PAF and reference ReLU output is too huge for training to correct.

To restrict the approximation error to the optimizable range of the training algorithm, Progressive Approximation (PA) applies the overall approximation error progressively instead applies all errors once. Specifically, PA replaces ReLU with PAF, one layer at a time, followed by fine-tuning all layers ahead of the replacement point in the neural network (Fig. 2)

In the first step of PA, the first ReLU is replaced with a PAF, and all layers ahead of the replacement, colored in blue in Fig. 3, are retrained until accuracy converges. Such a process is repeated until final convergence is achieved. PA effectively minimizes the deviation caused by the approximation error and thus effectively mitigates the accuracy degradation.

3.4 Alternate Training (AT)

After ReLU replacement by PAF, fine-tuning is required to recover the degraded accuracy. However, fine-tuning both convolution weights and PAF coefficients together could

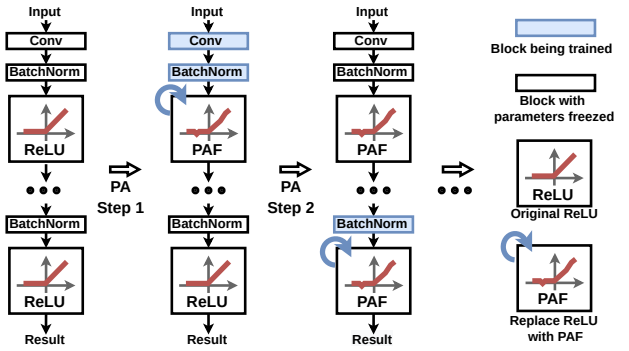

Figure 3. Demonstration of the Progressive Approximation (PA). In each step, a single ReLU layer gets replaced by the PAF followed by a fine-tuning for all other layers ahead of the replacement point.

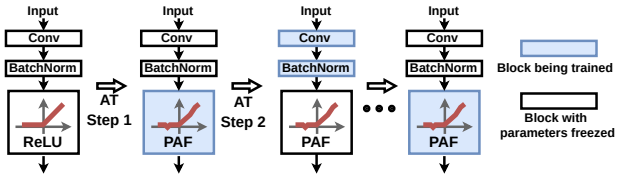

Figure 4. Alternate Training (AT) allows an interleave training between PAF coefficients and parameters of other layers to further reduce the accuracy degradation, where blue blocks are trained with other blocks freeze.

even lead to a higher accuracy drop, indicating a failure of convergence. The reason lies in the different impacts of convolution weights and coefficients of PAF on the final inference results. For example, a modification of value or even a pruned of value on convolution weights might not change the final inference results at all. While even a negligible change in coefficients of PAF could change every value in the activations layer. Intuitively, the training of coefficients of approximated polynomials should be decoupled from the training of convolution weights with even different hyperparameters.

Therefore, we propose Alternate Training to train PAF coefficients and parameters of other layers separately in an alternate manner. After each ReLU replacement, we will first train the PAF coefficients with other parameters fixed to optimize the PAF towards ReLU (Fig. 4). After a specific epoch threshold, PAF coefficients get frozen with training all other parameters. Such a training process intuitively is the same as training the neural network with another activation function instead of ReLU. The alternate training repeats until the accuracy finally converges, resulting in an alternating accuracy climbing. This approach effectively decouples the training of PAF coefficients from the training of other parameters and allows for different hyperparameters to train weights and PAF coefficients, resulting in improved accuracy and convergence.

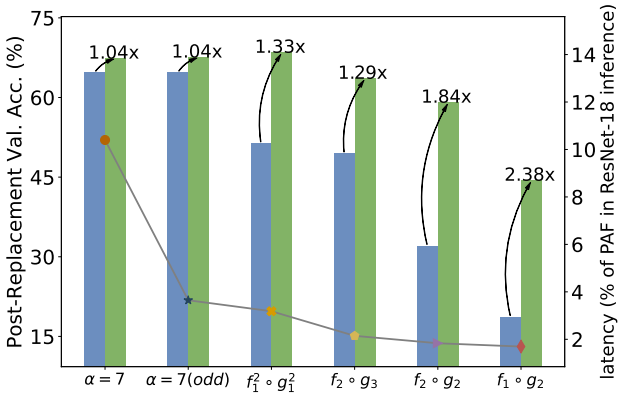

Figure 5. Coefficient Tuning (CT) compared to direct replacement.

4 Preliminary Evaluation

4.1 Coefficients Tuning Evaluation

Fig. 5 illustrates that Coefficients Tuning (CT) significantly improves overall validation accuracy by $1.04 \sim 2.38\times$ compared to the baseline shown in Tab. 2. CT yields greater benefits for polynomials with lower degrees. This is because polynomials with higher degrees have less overall approximated error across the entire input range, whereas low-degree polynomials have an insufficient capability of fitting the entire range, resulting in a significant decrease in accuracy. To mitigate this loss, CT focuses on fitting the high-probability region in the distribution, resulting in less approximated error.

4.1.1 Evaluation of Progressive Approximation

Progressive Approximation (PA) combines two techniques: (a) progressive replacement of ReLU with PAFs as opposed to direct replacement of all ReLU at once, and (b) progressive training of post-approximation model parameters compared to direct training of all parameters. This results in four different configurations, as shown in Fig. 6.

Among most PAFs with degrees ranging from 8 to 27, Progressive Approximation (PA) achieves the best overall accuracy because it allows gradual optimization of small deviations introduced by the approximation error. Furthermore, progressive training is more critical than progressive replacement, as the accuracy improvement between direct replacement with progressive training and progressive replacement with progressive training is similar. In some cases, direct replacement with progressive training even yields better validation accuracy, e.g. in the $f_1 \circ g_2$ configuration.

4.2 Ablation Study of All Proposed Techniques

To demonstrate the effectiveness of the proposed techniques, we conduct an ablation study with results presented in Tab. 3. Among all polynomials with different degrees, our proposed training techniques consistently improve accuracy over the baseline training strategy, by $1.03\times \sim 1.19\times$ for different degrees, rendering a $1.06 \sim 3.39\times$ accuracy compensation from direct replacement.

Table 3. Ablation study of all different proposed techniques

	$\alpha = 7$	$f_1^2 \circ g_1^2$	$f_2 \circ g_3$	$f_2 \circ g_2$	$f_1 \circ g_2$
direct replacement	64.70%	51.30%	49.40%	32.00%	18.60%
baseline	66.70%	64.30%	64.20%	58.30%	53.10%
CT	67.70%	68.60%	67.00%	66.50%	61.70%
AT	68.30%	65.20%	63.70%	60.50%	52.00%
PA	**68.40%**	65.60%	64.60%	60.20%	52.60%
PA + AT	67.40%	64.90%	64.60%	56.50%	47.10%
CT + PA	67.00%	68.20%	**67.60%**	65.90%	60.80%
CT + PA + AT	68.10%	**69.00%**	61.40%	**66.50%**	**63.10%**
Accuracy Improvement over direct replacement	1.06×	1.35×	1.37×	2.08×	3.39×
Accuracy Improvement over baseline	1.03×	1.07×	1.05×	1.14×	1.19×

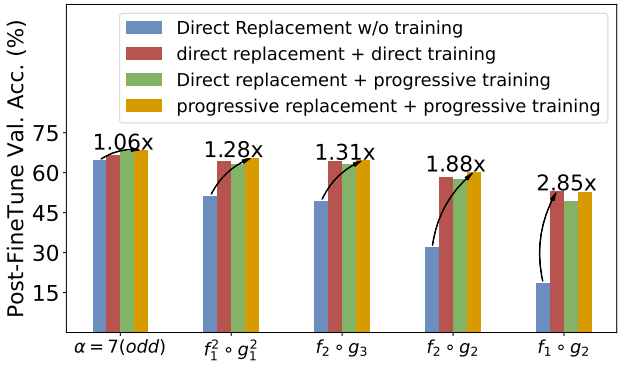

Figure 6. Progressive Approximation (PA) evaluation.

In summary, all PAFs with varying degrees create a natural tradeoff space between accuracy and latency. Our proposed approach allows us to identify the optimal point in this tradeoff space, a 12-degree PAF ($f_1^2 \circ g_1^2$) that achieves the same 69% post-approximation accuracy with 72% less latency compared to state-of-the-art 27-degree polynomials.

5 Conclusion

This paper demonstrates that the training of ML models with ReLU replaced with Polynomial Approximated Functions (PAF) is a fundamentally different problem. The typical training algorithm even leads to worse accuracy. To obtain the low-degree PAF without sacrificing accuracy, we propose a three-fold approach: (1) Coefficients Tuning to reduce approximation error between PAF and ReLU using profiled data distribution, (2) Progressive Approximation to divide and conquer the deviation introduced by approximation error, and (3) Alternate Training to improve post-approximation accuracy through decoupled training of PAF coefficients and other parameters. We explore the tradeoff space of PAF with variant degrees ranging from 8 to 27 and found that a 12-degree PAF yielded optimal results. Our evaluation of it on ResNet-18 (ImageNet 1k dataset) demonstrates a 69.0% overall post-approximated accuracy (0.3% accuracy degradation compared with pretrained ResNet-18) with 72% latency reduced compared to SotA 27-degree PAF. The proposed techniques potentially opens a new paradigm of obtaining PAF through ML training.

REFERENCES

Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., and Wernsing, J. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 201–210, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/gilad-bachrach16.html.

Kim, S., Kim, J., Kim, M. J., Jung, W., Kim, J., Rhu, M., and Ahn, J. H. Bts: An accelerator for bootstrappable fully homomorphic encryption. In *Proceedings of the 49th Annual International Symposium on Computer Architecture*, ISCA '22, pp. 711–725, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450386104. doi: 10.1145/3470496.3527415. URL https://doi.org/10.1145/3470496.3527415.

Lee, E., Lee, J.-W., No, J.-S., and Kim, Y.-S. Minimax approximation of sign function by composite polynomial for homomorphic comparison. *IEEE Transactions on Dependable and Secure Computing*, 19(6):3711–3727, 2022. doi: 10.1109/TDSC.2021.3105111.

Lee, J., Lee, E., Lee, J.-W., Kim, Y., Kim, Y.-S., and No, J.-S. Precise approximation of convolutional neural networks for homomorphically encrypted data. *ArXiv*, abs/2105.10879, 2021.

Lou, Q., Shen, Y., Jin, H., and Jiang, L. {SAFEN}et: A secure, accurate and fast neural network inference. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Cz3dbFm5u-.

Mateen, B. A., Liley, J., Denniston, A. K., Holmes, C. C., and Vollmer, S. J. Improving the quality of machine learning in health applications and clinical research. *Nature Machine Intelligence*, 2(10):554–556, 2020.

Raji, I. D. and Fried, G. About face: A survey of facial recognition evaluation. *CoRR*, abs/2102.00813, 2021. URL https://arxiv.org/abs/2102.00813.

Ran, R., Wang, W., Gang, Q., Yin, J., Xu, N., and Wen, W. CryptoGCN: Fast and scalable homomorphically encrypted graph convolutional network inference. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=VeQBBm1MmTZ.

Riazi, M. S., Laine, K., Pelton, B., and Dai, W. Heax: An architecture for computing on encrypted data. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '20, pp. 1295–1309, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371025. doi: 10.1145/3373376.3378523. URL https://doi.org/10.1145/3373376.3378523.

Zhang, Y., Wang, S., Zhang, X., Dong, J., Mao, X., Long, F., Wang, C., Zhou, D., Gao, M., and Sun, G. Pipezk: Accelerating zero-knowledge proof with a pipelined architecture. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*, pp. 416–428, 2021. doi: 10.1109/ISCA52012.2021.00040.