Towards Quantization-Native Zero-Knowledge Verification for INT4 Large Language Model Inference

George Chidera Akor [1], Love Allen Chijioke Ahakonye [2], Jae Min Lee [1], Dong-Seong Kim [1] *
[1] IT-Convergence Engineering, *Kumoh National Institute of Technology*, Gumi, South Korea
[2] ICT Convergence Research Center, *Kumoh National Institute of Technology*, Gumi, South Korea
* NSLab Co. Ltd., Gumi, South Korea, *Kumoh National Institute of Technology*, Gumi, South Korea
(georgeakor, loveahakonye, ljmpaul, dskim@kumoh.ac.kr)

***Abstract*—Zero-knowledge (ZK) verification of large language model (LLM) inference is bottlenecked by the sizes of lookup tables for non-linear operations. Current systems use tables with 2^{14} or more entries regardless of model precision. We observe that pre-quantized INT4 models constrain weights to 16 discrete values, a structure that existing ZK systems do not support. We present quantization-native ZK circuits with 16-entry weight lookups and 256-entry activation lookups implemented in Halo2. On a transformer feed-forward network (FFN) layer, we achieve 1,024× lookup reduction for weights and 64× for activations. We generate Halo2 Kate-Zaverucha-Goldberg (KZG) proofs with 3.44s of proving time, 26ms of verification, and 1.95KB of proof size.**

***Index Terms*—Halo2, KZG commitments, lookup tables, machine learning, transformer networks, zero-knowledge proofs.**

I. Introduction

Large language models have transformed numerous artificial intelligence (AI) applications, yet concerns surrounding their legitimacy and trustworthiness pose significant challenges [1]. Zero-knowledge proofs offer a promising approach where a prover can demonstrate correct model execution without revealing private weights or inputs [2]. Recent work has explored blockchain-based verification of machine learning (ML) inference to provide tamper-evident audit trails [3]. Current ZK-ML systems face a critical bottleneck in lookup tables for non-linear operations [4]. Systems such as EZKL and zkLLM use lookup tables with 2^{14} or more entries to handle activations and range checks [2], consuming significant memory and constraining scalability [5].

Zero-knowledge succinct non-interactive arguments of knowledge (zk-SNARKs) enable a prover to demonstrate knowledge of a witness satisfying a circuit without revealing the witness [4], with Halo2 providing PLONKish arithmetization and lookup arguments for efficient non-linear operations [6]. INT4 quantization reduces weights to 4-bit integers in {-8, . . . , +7}, with methods such as GPTQ [7] and AWQ [8] achieving near-lossless compression at 4× size reduction. Sun et al. demonstrated 13B parameter verification using tlookup for non-arithmetic operations [2], while the ZKML system [4]focuses on compiler optimizations; however, both use general-purpose lookup tables without quantization awareness. EZKL provides ONNX-to-Halo2 compilation but performs internal quantization without exploiting pre-quantized model structure [9]. Table I summarizes reported performance from prior ZKML systems on their respective benchmark models. These results are not directly comparable due to differing model architectures and scales.

TABLE I
Comparison with existing ZKML Systems

System	Model	Prove	Verify	Proof
zkLLM [2]	Llama-2-13B	803 s	3.95 s	200 KB
ZKML [4]	ResNet-18	52.9 s	12 ms	15.3 KB
EZKL [5]	CNN-Strided	69.8 s	–	479 KB
Ours (INT4)	FFN layer	3.44s	26ms	1.95KB

A key insight is observed that pre-quantized models have constrained value distributions. Models quantized with GPTQ [7] or AWQ [8] use INT4 weights with only 16 possible values. Current ZK systems ignore this structure, applying precision-agnostic verification to already-quantized models. Our contribution is quantization-native ZK circuits that exploit INT4/INT8 structure, addressing the gap between pre-quantized model distributions and precision-agnostic verification. We achieve 16-entry lookup tables for INT4 weights (1,024× reduction), 256-entry lookup tables for INT8 activations (64× reduction), and end-to-end verification of transformer FFN layers. Feed-Forward (FFN)layers.

II. Methodology

Fig. 1 illustrates our quantization-native verification pipeline. A pre-quantized INT4 model feeds into our Halo2-based circuit, which uses specialized lookup tables matched to the quantization precision.

A. Quantization-Native Circuit Design

Pre-quantized models have a known, fixed-value distribution, where INT4 weights take values in $\{-8, -7, \ldots, 7\}$, representing exactly 16 possibilities. We design our ZK circuit to exploit this structure through two specialized lookup tables. The INT4 weight lookup consists of a fixed table of 16 entries

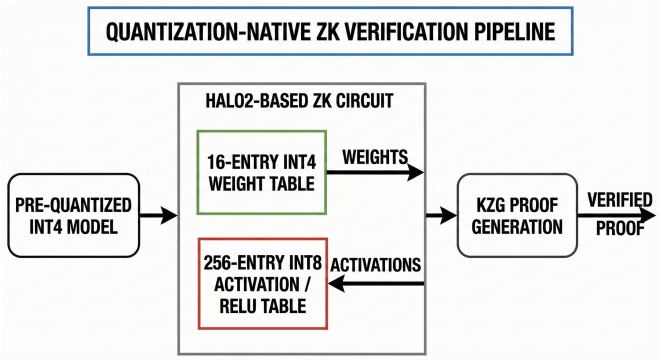

Fig. 1. Proposed Quantization-Native ZK Verification Pipeline. The architecture utilizes specialized Halo2 lookup tables matched to the model's numerical precision. By employing a 16-entry table for INT4 weights and a 256-entry table for INT8 activations, the system achieves significant reductions in circuit complexity and memory footprint compared to precision-agnostic approaches.

that map indices 0 through 15 to field elements representing -8 through $+7$. Each weight multiplication first verifies that the weight exists in this table via a lookup argument. The INT8 activation lookup uses a 256-entry table storing input-output pairs for all possible INT8 inputs to the rectified linear unit (ReLU) function. The two-column lookup verifies both input and output simultaneously.

B. FFN Layer Circuit

Our circuit implements a standard transformer FFN layer: a linear projection with INT4 weight verification, a ReLU activation via a 256-entry lookup, and an output projection with INT4 verification. For dimensions $d_{in} \to d_h \to d_{out}$, the circuit performs $(d_h \times d_{in} + d_{out} \times d_h)$ INT4 lookups and d_h ReLU lookups.

III. Results and Discussion

We evaluate on a scaled-down FFN layer (8→16→8) using Halo2 with BN256 curves and KZG commitments ($k = 12$). For fair comparison, we run EZKL on the identical model with three precision configurations. Table II presents results on the same FFN architecture.

TABLE II
Comparison with EZKL Baselines

Configuration	Lookup	Prove	Verify	Proof
EZKL (bits=7)	128	0.90 s	14.8 ms	19.22 KB
EZKL (bits=10)	1,024	1.98 s	22.7 ms	19.29 KB
EZKL (bits=14)	16,384	20.19 s	161.6 ms	19.22 KB
Ours (INT4)	**16**	3.44 s	26 ms	**1.95 KB**

Our quantization-native approach achieves 9.9× smaller proof size (1.95 KB versus 19.22 KB), which directly reduces on-chain verification costs. The lookup table is 8× smaller than EZKL's minimum configuration (16 versus 128 entries). While EZKL bits=7 achieves faster proving (0.90 s), it cannot exploit true INT4 structure; our 16-entry table precisely matches quantized weight distributions. Notably, EZKL prove time scales sharply with lookup size (22× slowdown from bits=7 to bits=14), whereas our approach maintains constant 16-entry tables regardless of model scale. For bandwidth-constrained deployments such as blockchain verification, the 10× proof size reduction offers substantial practical benefits.

IV. Conclusion

We demonstrated that quantization-native ZK circuits achieve dramatic efficiency improvements by exploiting pre-quantized model structure. Our 16-entry INT4 weight lookups and 256-entry activation lookups reduce table sizes by 64 to 1,024× compared to precision-agnostic approaches. Future work includes layer-streaming protocols for full model verification, on-chain verification via Ethereum smart contracts, and extension to attention mechanisms with INT4 key-value caches.

Acknowledgment

This work was partly supported by the Innovative Human Resource Development for Local Intellectualization program through the IITP grant funded by the Korea government (MSIT) (IITP-2025-RS-2020-II201612, 25%), by the Priority Research Centers Program through the NRF funded by the MEST (2018R1A6A1A03024003, 25%), and by the MSIT, Korea, under the ITRC support program (IITP-2025-RS-2024-00438430, 25%) and by the Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (RS-2025-25431637, 25%).

References

[1] Y. Huang, L. Sun, H. Wang, S. Wu, Q. Zhang, Y. Li, C. Gao, Y. Huang, W. Lyu, Y. Zhang *et al.*, "Trustllm: Trustworthiness in large language models," in *Proceedings of the 41st International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research. PMLR, 2024.

[2] H. Sun, J. Li, and H. Zhang, "zkllm: Zero knowledge proofs for large language models," in *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2024, pp. 4405–4419.

[3] G. C. Akor, L. A. C. Ahakonye, J. M. Lee, and D.-S. Kim, "Purechain-based zero-knowledge proofs for verifiable machine learning in industrial iot," in *Proceedings of the KICS Fall Conference*. Lahan Select Hotel, Gyeonggi, South Korea: Korean Institute of Communications and Information Sciences, 2025.

[4] B.-J. Chen, S. Waiwitlikhit, I. Stoica, and D. Kang, "Zkml: An optimizing system for ml inference in zero-knowledge proofs," in *Proceedings of the Nineteenth European Conference on Computer Systems (EuroSys'24)*. ACM, 2024, pp. 560–574.

[5] G. C. Akor, L. A. C. Ahakonye, J. M. Lee, and D.-S. Kim, "Benchmarking cnn components in ezkl: A layer-level analysis for evm-compatible deployment," in *Proceedings of the 8th International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. Tokyo, Japan: IEEE, 2026.

[6] Electric Coin Company and Privacy and Scaling Explorations, "The halo2 book," https://zcash.github.io/halo2, 2022.

[7] E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh, "Gptq: Accurate post-training quantization for generative pre-trained transformers," in *International Conference on Learning Representations (ICLR)*, 2022.

[8] J. Lin, J. Tang, H. Tang, S. Yang, W.-M. Chen, W.-C. Wang, G. Xiao, X. Dang, C. Gan, and S. Han, "Awq: Activation-aware weight quantization for on-device for llm compression and acceleration," in *Proceedings of Machine Learning and Systems (MLSys)*, vol. 6, 2024.

[9] Zkonduit, "Ezkl: Easy zero-knowledge machine learning," https://docs.ezkl.xyz, 2024.