

Late Breaking Results:

DAIQUIRI: Dynamic Quantization with Layer-wise Sensitivity Ranking for Hardware-Efficient LLMs

Tanha Tasfia*, Abu Kaisar Mohammad Masum*, Mehran Moghadam†, M. Hassan Najafi†, and Sercan Aygun*

*University of Louisiana at Lafayette, †Case Western Reserve University
{c00581680, c00591145, sercan.aygun}@louisiana.edu, {moghadam, najafi}@case.edu

Abstract—Large language models (LLMs) are widely used, but their high memory and compute demands make them difficult to deploy on low-power devices. Low-bit quantization reduces model size by converting full-precision weights into compact 2- or 3-bit formats, yet uniform low-bit quantization often forces a trade-off between efficiency and accuracy. In this work, we introduce DAIQUIRI, a dynamic low-bit quantization scheme that assigns bit-widths based on per-layer sensitivity. By using 2-bit precision for most layers and selectively applying 3-bit precision to more sensitive layers, DAIQUIRI maintains accuracy much higher than uniform 2-bit quantization while reducing energy, power, and latency compared to uniform 3-bit models. Experiments across multiple datasets and GPUs show that DAIQUIRI achieves accuracy close to the 3-bit baseline while using significantly fewer hardware resources, making it a practical and efficient solution for on-device LLM deployment.

Index Terms—Energy-efficient inference, GPU acceleration, Hardware-aware quantization, Large language models, Mixed-precision quantization.

I. INTRODUCTION

Large language models (LLMs) have found widespread use across many applications, yet their hardware-aware deployment remains constrained by substantial *memory* requirements and high *energy* consumption. *Low-bit weight quantization* offers a practical mechanism to alleviate these costs by reducing weight precision [1]. However, uniform low-bit quantization often introduces significant accuracy degradation. Recent approaches such as Activation-aware Weight Quantization (AWQ) [2], Generative Pre-Trained Vector Quantization (GPTVQ) [3], Flatness Matters for Quantization (FlatQuant) [4], and Quantization with Introspective Pruning (QuIP) [5] demonstrate that accuracy can be preserved at low bit-widths when the quantization process explicitly accounts for outlier weights. Nevertheless, these methods still adopt a uniform bit-width across all layers. This work proposes DAIQUIRI, a layer-sensitivity-aware *dynamic quantization scheme* built on Half-Quadratic Quantization (HQQ) [6] for LLMs. Uniform 2-bit quantization provides strong efficiency gains but suffers substantial accuracy degradation, while 3-bit precision preserves accuracy at a significantly higher energy and memory cost. Fig. 1 illustrates this trade-off across three datasets on both the NVIDIA TITAN RTX and RTX PRO 6000 Blackwell GPUs. Dynamic HQQ consistently achieves lower energy consumption compared to uniform 3-bit quantization. By allocating higher precision only to layers identified as sensitive through a small calibration subset and maintaining lower precision for more robust layers, the proposed *greedy bit-allocation strategy* satisfies a global average-bit constraint while retaining higher accuracy close to the 3-bit baseline at substantially reduced energy and memory cost.

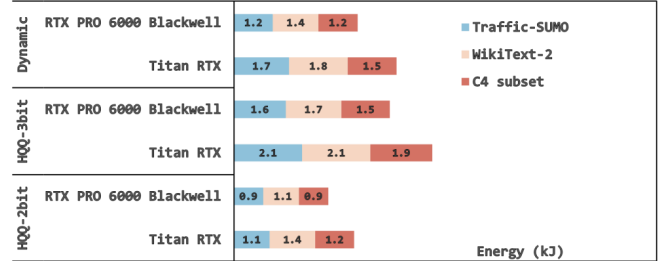


Fig. 1. Energy consumption of HQQ-2bit, HQQ-3bit, and the proposed Dynamic HQQ per unit task across three datasets on NVIDIA TITAN RTX and NVIDIA RTX PRO 6000 Blackwell GPUs.

II. METHODOLOGY

In this section, we describe DAIQUIRI, our layer-sensitivity-aware dynamic quantization scheme for LLMs. We begin by formulating low-bit quantization as a constrained bit-allocation problem governed by a global average-bit budget. We then introduce a calibration-based sensitivity metric that ranks transformer layers according to their impact on task loss when quantized to lower precision. Finally, we present a greedy allocation algorithm that selectively upgrades only the most sensitive layers from 2-bit to 3-bit precision.

Our target model is an instruction-tuned Llama-3.2-1B, whose transformer linear layers are quantized using the HQQ framework [6]. For evaluation, we consider three configurations across three datasets: (i) *HQQ-2bit*, a uniform 2-bit model; (ii) *HQQ-3bit*, a uniform 3-bit model; and (iii) Dynamic HQQ (DAIQUIRI), a mixed-precision variant with an average of 2.6 bits per weight and per-layer bit-widths in $\{2, 3\}$. Tokenization, sequence length, and optimization hyperparameters are held constant across all experiments.

During training, the quantized backbone remains frozen, and only lightweight LoRA adapters are applied to the attention and feed-forward projections, preserving the efficiency benefits of low-bit inference. Training is conducted separately on three datasets: WikiText-2 (language modeling) [7], a C4 subset (long-form text) [8], and a SUMO traffic-scenario dataset (code generation) [9]. Each example is formatted as an *instruction-response* sequence: $x-y_{1:T}$ (with T varying across datasets/examples but capped by the maximum sequence length). Models are optimized using the standard autoregressive cross-entropy objective $\ell(x, y; \theta) = -\frac{1}{T} \sum_{t=1}^T \log p_{\theta}(y_t | x, y_{<t})$, where x denotes the input instruction (or prompt), $y_{1:T}$ is the target response token sequence, and θ are the model parameters. The empirical loss $L(\theta; D) = \frac{1}{|D|} \sum_{(x,y) \in D} \ell(x, y; \theta)$ is minimized independently for each dataset D and each quantization configuration.

DAIQUIRI follows the three-stage pipeline illustrated in

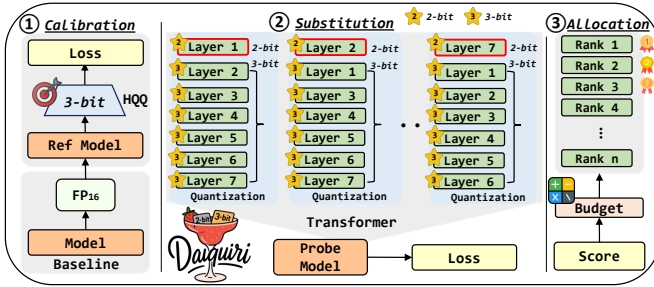


Fig. 2. **DAIQUIRI**: Dynamic quantization pipeline. A 3-bit reference model is first calibrated; then each transformer layer is temporarily switched to 2-bit to measure its sensitivity; finally, layers are ranked by sensitivity and assigned 2- or 3-bit precision under an *average-bit* budget.

TABLE I

PERPLEXITY (PPL) ON WIKITEXT-2 AND A HELD-OUT C4 SUBSET

Method	Average bits	WikiText-2 PPL ↓	C4-subset PPL ↓
HQQ-3bit (uniform)	3.0	7.13	9.75
HQQ-2bit (uniform)	2.0	13.29	14.90
DAIQUIRI	2.6	7.18	9.81

Fig. 2: ① *Calibration*, ② *Substitution*, and ③ *Allocation*. In the *Calibration* stage, a 3-bit HQQ reference model θ^{ref} is evaluated on a small calibration subset $\mathcal{C}_D \subset D$ to obtain the baseline loss L_{ref} . In the *Substitution* stage, the quantization sensitivity of each transformer layer is measured; For each layer ℓ , a probe model $\theta^{(\ell,b)}$ is formed by temporarily assigning 2-bit precision to that layer while all other layers remain at 3-bit. The resulting calibration loss $L_{\ell,b}(D)$ produces a sensitivity score $\Delta_{\ell,b}(D) = L_{\ell,b}(D) - L_{\text{ref}}(D)$, where a larger $\Delta_{\ell,2}(D)$ indicates that layer ℓ is less tolerant to aggressive 2-bit quantization. In the *Allocation* stage, layers are ranked by their sensitivity scores, and bit-widths are assigned under an average-bit constraint. Let layer ℓ contain N_ℓ weights and be assigned bit-width $k_\ell \in \{2, 3\}$. The weighted average precision is $\bar{k} = \frac{\sum_\ell k_\ell N_\ell}{\sum_\ell N_\ell}$. All layers are initialized with $k_\ell = 2$, and the most sensitive layers are progressively upgraded to 3-bit until the target average of \bar{k} is reached. The resulting dataset-specific bit-width map is encoded into HQQ’s dynamic configuration and applied consistently during fine-tuning and evaluation.

III. EXPERIMENTAL EVALUATION

We fine-tuned Llama-3.2-1B separately on each dataset and evaluated performance on the corresponding test split. Three quantization settings, uniform 2-bit, uniform 3-bit, and **DAIQUIRI**, were evaluated under identical decoding conditions. During inference, we recorded latency (mean wall-clock time per prompt), average GPU power, and energy per prompt. We measured accuracy using perplexity (PPL) for WikiText-2 and the C4 subset, and pass@k for the Traffic-SUMO code-generation task. All experiments were performed on two GPUs representing different hardware tiers.

Table I presents the perplexity results, while Table II shows the pass@k performance for Traffic-SUMO. Table III reports average power measurements, and Table IV reports end-to-end latency results across datasets and GPUs. As expected, uniform 2-bit quantization yields the lowest accuracy, whereas uniform 3-bit quantization achieves the highest. **DAIQUIRI** offers a middle ground, achieving accuracy much closer to the 3-bit baseline while adhering to a lower bit-width budget. In terms

TABLE II

PASS@K PERFORMANCE ON TRAFFIC-SUMO CODE GENERATION TASK

Method	pass@1	pass@2	pass@3	pass@4
HQQ-3bit	0.00	0.12	0.30	0.42
HQQ-2bit	0.00	0.00	0.12	0.25
DAIQUIRI	0.00	0.09	0.28	0.40

TABLE III

AVERAGE POWER CONSUMPTION (kW) ACROSS GPUS AND DATASETS

Method	Dataset	TITAN RTX	Blackwell
HQQ-3bit	Traffic-SUMO	0.26	0.58
	WikiText-2	0.24	0.58
	C4 subset	0.20	0.48
HQQ-2bit	Traffic-SUMO	0.22	0.50
	WikiText-2	0.24	0.55
	C4 subset	0.18	0.42
DAIQUIRI	Traffic-SUMO	0.24	0.54
	WikiText-2	0.24	0.56
	C4 subset	0.19	0.45

TABLE IV

LATENCY (S) ACROSS GPUS AND DATASETS

Method	Dataset	TITAN RTX	Blackwell
HQQ-3bit	Traffic-SUMO	8.1	2.7
	WikiText-2	9.0	3.0
	C4 subset	9.6	3.2
HQQ-2bit	Traffic-SUMO	5.2	1.7
	WikiText-2	6.0	2.0
	C4 subset	6.5	2.2
DAIQUIRI	Traffic-SUMO	6.9	2.3
	WikiText-2	7.5	2.5
	C4 subset	8.1	2.7

TABLE V

COMPARISON WITH OTHER LOW-BIT QUANTIZATION METHODS [10]

Method	WikiText-2 PPL ↓	C4 PPL ↓
GPTVQ [3]	6.25	7.97
AWQ [2]	6.24	7.84
QuIP [5]	6.80	7.75
LLM-MQ [11]	7.16	8.94
DAIQUIRI	6.22	7.74

of latency, Blackwell GPU consistently outperforms TITAN RTX across all quantization settings owing to its higher inference throughput. Table V reports 3-bit and mixed-precision results on LLaMA-2-7B, including both **DAIQUIRI** (mixed precision) and competitive 3-bit baselines, evaluated at a larger model scale to enable direct comparison with prior low-bit quantization methods.

IV. CONCLUSION

Dynamic quantization offers a balanced trade-off between accuracy and efficiency by assigning bit widths based on layer sensitivity. Our **DAIQUIRI** scheme achieves accuracy close to 3-bit models while reducing power and latency, making LLMs more suitable for deployment on resource-constrained hardware.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation grant No. 2019511, 2339701, NASA grant 80NSSC25C0335, Vernon & Ruby Langlinais Non-Endowed Res. Fund, Lockheed Martin Corporation Endowed Professorship, award from NASA, and generous gifts from NVIDIA.

REFERENCES

- [1] K. Egashira, M. Vero, R. Staab, J. He, and M. Vechev, "Exploiting llm quantization," in *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, Eds., vol. 37. Curran Associates, Inc., 2024, pp. 41 709–41 732. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/file/496720b3c860111b95ac8634349dcc88-Paper-Conference.pdf
- [2] 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 llm compression and acceleration," in *Proceedings of Machine Learning and Systems*, P. Gibbons, G. Pekhimenko, and C. D. Sa, Eds., vol. 6, 2024, pp. 87–100. [Online]. Available: https://proceedings.mlsys.org/paper_files/paper/2024/file/42a452cbafa9dd64e9ba4aa95cc1ef21-Paper-Conference.pdf
- [3] M. van Baalen, A. Kuzmin, I. Koryakovskiy, M. Nagel, P. Couperus, C. Bastoul, E. Mahurin, T. Blankevoort, and P. Whatmough, "Gptvq: The blessing of dimensionality for llm quantization," 2025. [Online]. Available: <https://arxiv.org/abs/2402.15319>
- [4] Y. Sun, R. Liu, H. Bai, H. Bao, K. Zhao, Y. Li, J. Hu, X. Yu, L. Hou, C. Yuan, X. Jiang, W. Liu, and J. Yao, "Flatquant: Flatness matters for llm quantization," 2025. [Online]. Available: <https://arxiv.org/abs/2410.09426>
- [5] A. Tseng, J. Chee, Q. Sun, V. Kuleshov, and C. D. Sa, "Quip#: Even better llm quantization with hadamard incoherence and lattice codebooks," 2024. [Online]. Available: <https://arxiv.org/abs/2402.04396>
- [6] H. Badri and A. Shaji, "Half-quadratic quantization of large machine learning models," November 2023. [Online]. Available: https://dropbox.github.io/hqq_blog/
- [7] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," 2016. [Online]. Available: <https://arxiv.org/abs/1609.07843>
- [8] J. Dodge, M. Sap, A. Marasovic, W. Agnew, G. Ilharco, D. Groeneveld, M. Mitchell, and M. Gardner, "Documenting large webtext corpora: A case study on the colossal clean crawled corpus," 2021. [Online]. Available: <https://arxiv.org/abs/2104.08758>
- [9] R. Pradhananga, S. Williams, S. Aygun, L. Chen, Y. Tu, W. Crow, S. Aakur, and N. Tzeng, "Digital twin-aided municipal traffic control," in *Proceedings of the SUMO User Conference 2025 (SUMO Conference Proceedings)*, vol. 6, Germany, 2025, pp. 149–161.
- [10] Z. Chen, B. Xie, J. Li, and C. Shen, "Channel-wise mixed-precision quantization for large language models," 2025. [Online]. Available: <https://arxiv.org/abs/2410.13056>
- [11] S. Li, X. Ning, K. Hong, T. Liu, L. Wang, X. Li, K. Zhong, G. Dai, H. Yang, and Y. Wang, "Llm-mq: Mixed-precision quantization for efficient llm deployment," in *The Efficient Natural Language and Speech Processing Workshop with NeurIPS*, 2023.