Kashf Journal of Multidisciplinary Research
Vol: 03 - Issue 5 (2026)
P-ISSN: 3007-1992 E-ISSN: 3007-200X
https://kjmr.com.pk

ENERGY-EFFICIENT NLP THROUGH TINY-MODEL DISTILLATION, PRUNING, AND QUANTIZED INFERENCE

***Hamadullah Bijarani*[1]**

[1]*Dept. of Computer Systems Engineering Sukkur IBA University Sukkur, Pakistan*

Corresponding author email: *hamadullah.becsef23@iba-suk.edu.pk*

Article Info

Abstract

Transformer models achieve state-of-the-art NLP performance but face deployment constraints due to high in-ference energy costs. Numerous techniques have been proposed to optimize model operations. Due to limitations of hardware and time, we analyzed a few of them, which are knowledge distillation, unstructured pruning, and quantization within a uni-fied pipeline for BERT and TinyBERT. Results demonstrate that the Distilled Student serves as the dominant baseline, offering the lowest energy per sample among all variants. Conversely, while post-training quantization and pruning reduce model size, they introduce significant kernel overheads on GPUs. Notably, combining INT8 quantization with aggressive pruning results in severe latency penalties, causing up to a 4x increase in energy consumption compared to the standard Student. These findings underscore the critical disparity between theoretical compression metrics and hardware-realized energy efficiency.

Keywords: Responsible AI (RAI, efficient AI, distillation, model compression, BERT, TinyBERT, Artificial Neural Network (ANN), neural network prunning, energy-efficient AI

1. INTRODUCTION

A. Background and Motivation

Transformer architecture-basedmodels, particularly BERT [1], have become the de facto standard for Natural Language Processing (NLP). These gains, however, come with quite large computational and energy costs, especially during infer-ence at scale [2]. As NLP models are increasingly deployed in continuous services, inference efficiency has emerged as a primary bottleneck. This shift motivates the principles of "Green AI," which advocate optimizing models for energy consumption and en-vironmental impact [3]. Importantly, the carbon footprint of sustained inference can match that of training for widely deployed models queried over long lifecycles [4]. In resource-constrained environments such as edge devices or consumers, energy efficiency is a hard constraint governed by battery capacity and thermal limits.

B. Problem Statement

Despite extensive research on model compression, a per-sistent gap exists between algorithmic efficiency gains and real-world energy savings. Most prior work relies on proxy metrics such as parameter counts or FLOPs. However, as described by the Roofline model [5], hardware-level energy consumption depends heavily on memory access patterns and kernel efficiency, factors not captured by these proxies [6]. From a systems perspective, a reduction in FLOPs may not translate to lower energy if execution becomes memory-bound

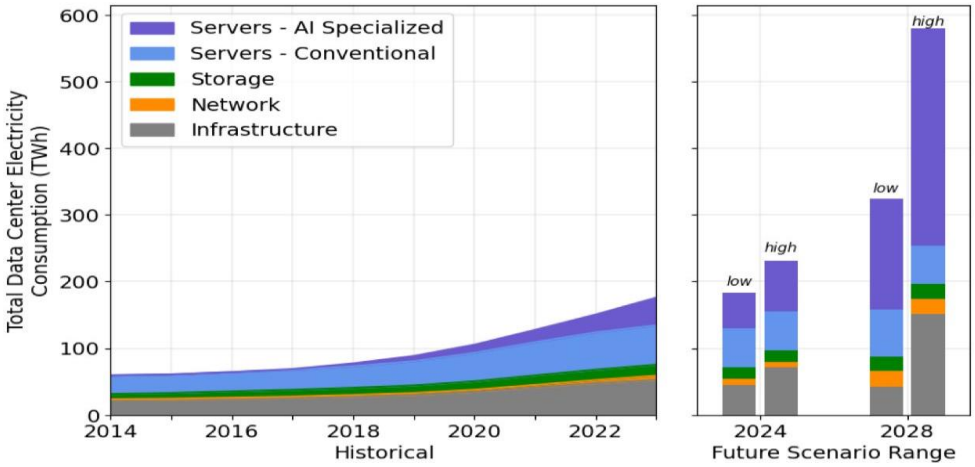

Figure 1: Historical and projected data-center electricity consumption highlight-ing growth driven by AI workloads

or incurs additional kernel overhead. This disconnect compli-cates deployment decisions and motivates direct, hardware-level measurement of inference energy.

C. Contributions

We bridge the gap between algorithmic compression and hardware reality through the following contributions:

1) We use a single, experimental framework to evaluate the cumulative energy impact of knowledge distillation, unstructured pruning, and low-bit quantization for a student model.

2) We report direct, GPU power measurements obtained under controlled inference conditions, verifying data against multiple runs and raw benchmarks.

3) We provide practical, hardware-grounded insights into the efficacy of compression techniques, distinguishing between theoretical reductions and tangible real-world energy savings.

II. Related Work

A. Model Compression

Knowledge distillation [7] transfers knowledge from a large teacher to a smaller student, as seen in TinyBERT [8] and DistilBERT [9]. Pruning removes redundant weights to induce sparsity [10]. The Lottery Ticket Hypothesis [11] suggests that dense networks contain sparse, matching subnetworks capableof high accuracy. Recent works like HAP-E focus on Hessian-aware structured pruning for LLMs [17], while others explore block-sparse pruning [18]. Quantization reduces numerical precision. While INT8 is industry standard [12], but recent advances explore INT4 and activation quantization techniques like SmoothQuant [19] and GPTQ [20] for larger models.

B. Energy Efficiency

Recent surveys highlight the discrepancy between theo- retical FLOPs and actual energy usage [16]. Many studies implicitly assume that reductons in model size or latency will proportionally reduce energy consumption, an assumption that does not always hold in practice due to hardware utilization nuances [6].

III . METHODOLOGY

A. Compression Pipeline

We employ a multi-stage compression pipeline (Fig. 2). A large Teacher model distills knowledge to a compact Student, which is subsequently pruned and quantized.

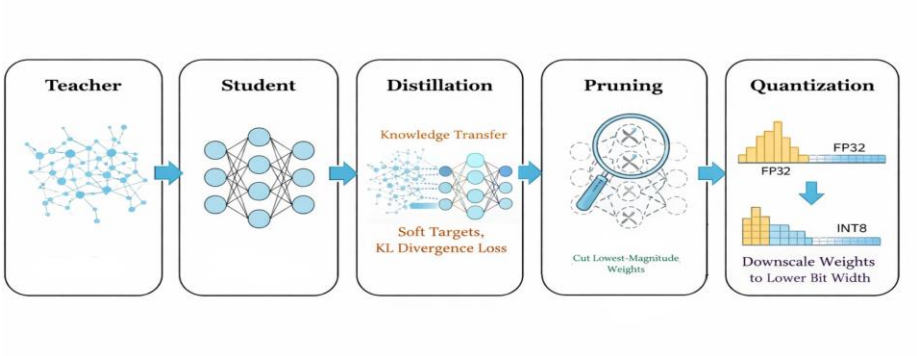

Figure 2:Proposed pipeline: Teacher → Distillation → Student → Pruning →

B. Knowledge Distillation

We distill knowledge from a BERT-base teacher to a Tiny-BERT student. To improve convergence, we employ interme-diate layer distillation using Mean Squared Error (MSE) on hidden states. The student hidden states H_S are mapped to teacher states H_T via a learnable matrix W_h:

$$L_{hid} = \sum_{i=1} MSE(H_S^{(i)} W_h, H_T^{(i)}) \quad (1)$$

The prediction layer loss combines Cross-Entropy (L_{CE}) and KL-Divergence:

$$L_{pred} = \alpha L_{CE} + \beta T^2 L_{KL}(p^S/T, p^T/T) \quad (2)$$

where p represents the logits.

C. Unstructured Pruning

We apply magnitude-based pruning [10]. Formally, we formulate the optimization problem to minimize loss subject to an L_0 norm constraint on the weights:

$$\min_W L(W; D) \quad \text{s.t.} \quad ||W||_0 \leq \kappa \quad (3)$$

The pruned weight matrix W' is defined using a binary mask M :

$$W' = W \odot M, \qquad M_{ij} = I(|W_{ij}| > \theta) \quad (4)$$

D. Quantization (INT8 & INT4)

We conduct an evaluation of few post-training quantization techniques. Here, a floating-point tensor x is mapped to integer q via scale factor S and zero-point Z:

$$q = \text{clip}\left(\text{round}\left(x/S + Z\right), q_{min}, q_{max}\right) \quad (5)$$

We use dynamic INT8 quantization [12] and INT4 weight quantization via the bitsandbytes library [13].

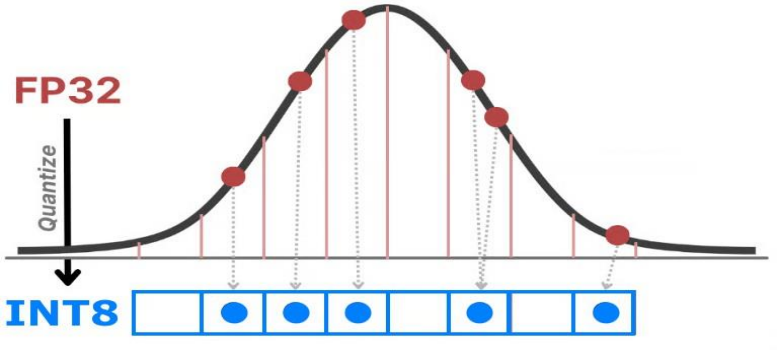

Fig. 3. Mapping continuous FP32 values to discrete INT8 buckets

A. Energy Measurement Protocol

All of the experiments are conducted on a single NVIDIA Tesla T4 GPU. The total energy consumption is obtained by numerically integrating power measurements over the infer-ence duration using the trapezoidal rule:

$$E = \sum_{t=1}^{N} \frac{P_t + P_{t+1}}{2} \cdot \Delta t \qquad (6)$$

We also consider the Energy Delay Product (EDP) to penalize slow, low power models: $EDP = E \times$ Latency.

VI. Experimental Setup

Experiments were conducted on an NVIDIA Tesla T4 GPU (16 GB GDDR6) using PyTorch and HuggingFace Transform-ers. Inference was performed with a fixed batch size of 16 on the SST-2 dataset [15]. All models underwent warm-up passes to stabilize CUDA kernels before measurement. The reported results have been verified against raw benchmark logs to ensure data integrity.

V. RESULTS

A. Accuracy Analysis

Table I presents the classification accuracy. The distilled stu-dent establishes the performance baseline. Pruning up to 30% shows negligible loss. Interestingly, our verified results show that INT8 quantization combined with 30% pruning retains surprisingly high accuracy (87.4%), only a 1.6% drop from the student baseline, contradicting the expectation of severe degradation. However, INT4 quantization remains sensitive to sparsity.

Table 1: ACCURACY RESULTS ON SST-2 (VERIFIED)

Model Variant	Acc (%)	Drop (%)
Teacher (BERT-base)	91.0	–
Distilled Student	89.0	0.0
+ Pruning (20%)	88.8	0.2
+ Pruning (30%)	88.1	0.9
+ Pruning (40%)	50.9	38.1
+ INT8 Quantization	87.7	1.3

Model Variant	Acc (%)	Drop (%)
++ Pruning (20%)	88.4	0.6
++ Pruning (30%)	87.4	1.6
++ Pruning (40%)	50.9	38.1
++ Pruning (20%)	84.7	4.3
++ Pruning (30%)	68.1	20.9
++ Pruning (40%)	50.9	38.1

B. Latency and Throughput

Table II details inference speed. The Distilled Student sets the speed baseline at 8.86 ms. Unstructured pruning maintains this baseline performance. However, INT8 and INT4 quan-tization introduced latency bottlenecks. Critically, combining INT8 with 30% and 40% pruning caused a severe degradation in latency (up to 96 ms), which is 11x slower than the standard Student.

Table 2:INFERENCE LATENCY (BATCH TIME) AND THROUGHPUT (VERIFIED)

Model Variant	Batch Latency (ms)	Throughput (samples/s)
Teacher (BERT-base)	133.26	126.5
Distilled Student	8.86	1,817.9
+ Pruning (20%)	8.94	1,755.8
+ Pruning (30%)	8.84	1,763.9
+ Pruning (40%)	8.87	1,814.0
+ INT8 Quantization	58.15	257.8
++ Pruning (20%)	64.47	212.7
++ Pruning (30%)	96.30	207.2
++ Pruning (40%)	92.81	206.1
+ INT4 Quantization	31.45	492.3
++ Pruning (20%)	31.88	475.5

Model Variant	Batch Latency (ms)	Throughput (samples/s)
++ Pruning (30%)	32.35	471.5
++ Pruning (40%)	32.05	479.9

C. Energy Consumption

Table III summarizes the energy metrics. The Distilled Stu-dent configuration demonstrates the highest energy efficiency, requiring just 0.05 J per sample. While INT8 quantization contributed to lowering average power draw, the increased latency (especially when combined with pruning) resulted in significantly higher total energy per sample. The INT8+30% Pruned model, despite its high accuracy, consumes 0.24 J/sample, which is 4.8 times more energy than the standard Student due to the 96ms latency penalty.

D. The Pareto Frontier

Figure 5 illustrates the trade-off. The FP32 Student lies on the optimal frontier. Despite the improved accuracy of the

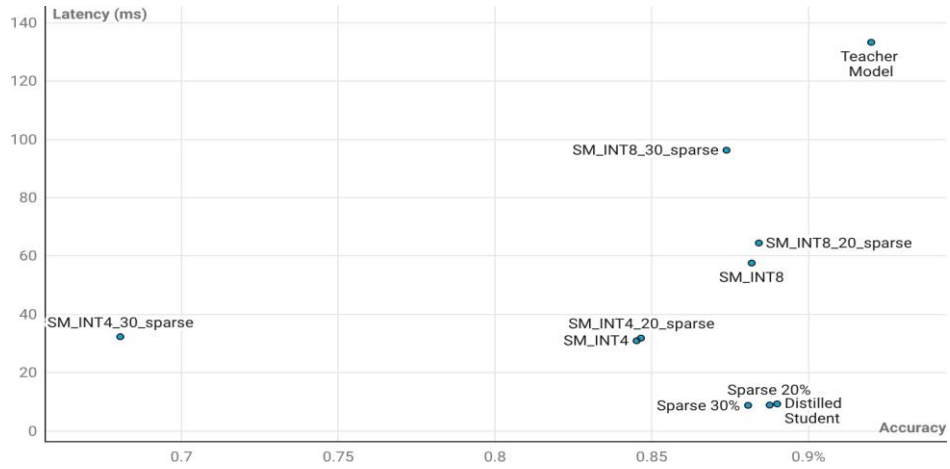

Figure 4: Decomposition of inference efficiency into average power draw (W) and latency (ms) across model variants.

Table 3:INFERENCE ENERGY CONSUMPTION (VERIFIED)

Model Variant	Power (W)	Energy/Sample (J)	Size (MB)
Teacher (BERT)	69.80	0.5800	417.7
Distilled Student	67.53	0.0500	54.8
+ Pruning (20%)	66.78	0.0400	54.8
+ Pruning (30%)	61.88	0.0500	54.8
+ Pruning (40%)	69.13	0.0500	54.8
+ INT8 Quant	38.09	0.1400	41.5
+ + Pruning (20%)	36.13	0.1453	41.5
+ + Pruning (30%)	33.78	0.2035	41.5
+ + Pruning (40%)	32.68	0.1892	41.5
+ INT4 Quant	66.64	0.1400	39.3
+ + Pruning (20%)	60.56	0.1206	39.3
+ + Pruning (30%)	59.33	0.1199	39.3
+ + Pruning (40%)	61.46	0.1235	39.3

INT8+Pruning models, their increased energy consumption (up to 0.24 J) disqualifies them from the Pareto frontier compared to the efficient Distilled Student (0.05 J). frontier compared to the efficient Distilled Student (0.05 J).

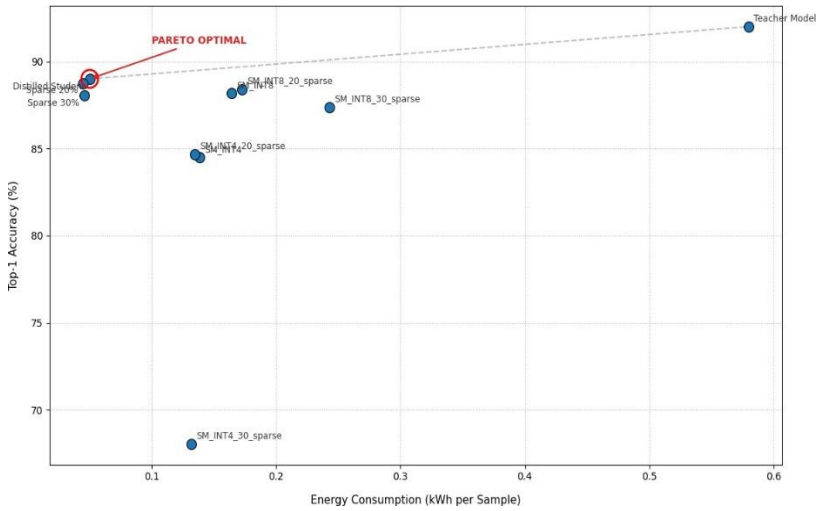

Figure 5: Accuracy (Y-axis) vs. Energy Consumption (X-axis). The Distilled Student represents the ”Sweet Spot” on the Pareto frontier

VI.Discussion and Limitations

The results highlight that the Distilled Student is the most consistent strategy for general purpose GPUs. Unstructured pruning requires sparse-aware hardware to translate FLOP re-ductions into energy savings. The verification process revealed a critical anomaly: combining INT8 with high-sparsity pruning (30%+) leads to severe kernel overhead, increasing latency by 11x compared to the standard Student. This contradicts theoretical expectations and emphasizes the gap between soft-ware implementations and hardware capabilities. Future work should explore other parameter-efficient finetuning techniques like LoRA [14] and structured sparsity.

VII . Conclusion

We presented a hardware-based energy evaluation of com-pressed NLP models, verified against multiple runs, and raw benchmark data. The Distilled Student emerged as the Pareto-optimal baseline, delivering the lowest energy consumption (0.05 J/sample) with minimal accuracy loss. In contrast, stan-dard pruning and the quantization pipelines failed to match this efficiency. Specifically, INT8+Pruning, while accuracy-friendly, suffers from prohibitive latency costs, increasing energy consumption by up to 4.8x compared to the standard Student. This necessitates the direct hardware measurement for accurate energy profiling.

REFERENCES

J. Devlin et al., "BERT: Pre-training of deep bidirectional transformers for language understanding," Proc. NAACL, 2019.

Strubell et al., "Energy and policy considerations for deep learning in NLP," Proc. ACL, 2019.

R. Schwartz et al., "Green AI," CACM, vol. 63, no. 12, 2020.

Patterson et al., "Carbon emissions and large neural network training," arXiv:2104.10350, 2021.

S. Williams et al., "Roofline: an insightful visual performance model for multicore architectures," CACM, vol. 52, 2009.

Yoon, J. Mun, and K.-S. Min, "Comparative study on energy con-sumption of neural networks by scaling of weight-memory energy ver-sus computing energy for implementing low-power edge intelligence," Electronics, vol. 14, no. 13, p. 2718, 2025.

E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv preprint arXiv:1503.02531, 2015.

X. Jiao et al., "TinyBERT: Distilling BERT for natural language under-standing," Findings of EMNLP, 2020.

V. Sanh et al., "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," arXiv preprint arXiv:1910.01108, 2019.

S. Han et al., "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," ICLR, 2016.

J. Frankle and M. Carbin, "The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks," ICLR, 2019.

Jacob et al., "Quantization and training of neural networks for efficient integer-arithmetic-only inference," CVPR, 2018.

T. Dettmers et al., "LLM.int8(): 8-bit matrix multiplication for trans-formers at scale," NeurIPS, 2022.

J. Hu et al., "LoRA: Low-Rank Adaptation of Large Language Models," ICLR, 2022.

R. Socher et al., "Recursive deep models for semantic compositionality over a sentiment treebank," EMNLP, 2013.

K. T. Chitty-Venkata et al., "A Survey of Techniques for Optimizing Transformer Inference," J. Syst. Archit., 2023.

Adepu et al., "HAP-E: Hessian-Aware Structured Pruning of LLMs for Efficient Inference," arXiv preprint arXiv:2405.14737, 2024.

Lagunas et al., "Block pruning for faster transformers," EMNLP, 2021.

Xiao et al., "SmoothQuant: Accurate and efficient post-training quantization for large language models," ICML, 2023.

Frantar et al., "GPTQ: Accurate post-training quantization for gener-ative pre-trained transformers," ICLR, 2023.