Energy–Accuracy Trade-offs in Transformer-Based NLP Models: A Unified Benchmarking Study

Thibaud Clement

Department of Computer Science, Stanford University

Problem

Transformer-based NLP models achieve strong **performance** but require substantial computational and energy **resources**:

- As models scale, energy consumption during training and inference is becoming a major **economic and environmental concern**.
- Many techniques have been proposed to **improve efficiency**, but they are usually evaluated in isolation.
- As a result, it remains unclear how these techniques compare under consistent experimental conditions and what **energy–accuracy trade-offs** they produce in practice.
- However, practitioners deploying NLP systems must **decide which efficiency strategies** to use under different computational and energy constraints.

Key research question: How do different efficiency strategies shift the energy–accuracy trade-off of transformer-based NLP models under consistent experimental conditions?

Background

A growing body of work has examined the computational and environmental costs of modern machine learning systems, motivating research on more energy-efficient NLP models:

- Energy and sustainability in ML
  - A single training run can require substantial energy resources, in the order of hundreds of megawatt-hours.
  - The "Green AI" paradigm advocates treating efficiency as a primary evaluation criterion alongside accuracy.
- Computational scaling constraints
  - Improving the efficiency of transformer models is challenging because computational cost scales with both model size and input sequence length.
  - Many components of the network contribute to predictive performance, making it difficult to reduce computational cost without sacrificing accuracy.
- Architectural and inference-time techniques
  - Methods such as knowledge distillation, parameter sharing, layer freezing, and pruning aim to reduce model size or complexity while preserving predictive performance.
  - Reduced precision computation, sequence length truncation, token pruning, and early exiting aim to reduce the amount of computation performed during inference.

Research gap: Efficiency techniques are typically evaluated in isolation, making it difficult to understand how different strategies compare or interact under consistent experimental conditions.

Methods

Model architectures and baselines

Model	Strategy	Description
BERT	Baseline	12-layer transformer (110M parameters) used as reference
DistilBERT	Distillation	Compressed, faster derivative of BERT
BERT-Freeze6	Layer freezing	Lower 6 layers frozen during fine-tuning
BERT-Edge	Layer freezing	Only first and last layers updated
BERT-Prune50	Magnitude pruning	Approximately 50% of parameters removed
BERT-EE	Early exit	Modified BERT to support early-exit inference

Table 1. BERT-derived architectures evaluated in this study.

Inference-time efficiency techniques

Technique	Effect
Sequence length truncation	Limits maximum input tokens, reducing quadratic self-attention cost
Reduced precision	Performs inference using lower numerical precision
Token pruning	Retains only the most relevant tokens for later layers
Early exit	Stops inference once confidence exceeds a threshold

Table 2. Inference-time reductions evaluated in this study.

Measurement framework

We measure the energy consumption of training and inference by sampling GPU power during execution and integrating power over time. With $P(t)$ denoting instantaneous GPU power in watts at time t and T the total runtime of a given operation, the total energy consumption E, expressed in joules, is computed as:

$$E = \int_0^T P(t), dt.$$

Power measurements are obtained via periodic GPU telemetry queries using `nvidia-smi`. Energy metrics are reported as energy per example (J/example) to enable comparison across configurations.

Experiments

Experimental setup

Component	Details
Task	SQuAD v2 extractive question answering. Models receive a context paragraph and question and predict an answer span or a null answer.
Evaluation	Accuracy: Exact Match (EM) and token-level F1 (F1 used in energy–accuracy plots). Efficiency: inference energy per example and total training energy.
Experimental axes	Architectural variants (Table 3) and inference-time techniques (Table 2). Sweeps include sequence length $\{384, 320, 288, 256, 224, 192\}$, precision $\{FP32, FP16, BF16\}$, token pruning ratios $\{1.0, 0.8, 0.6, 0.4, 0.3\}$, and early-exit thresholds $\{1.01, 0.90, 0.80, 0.70, 0.60, 0.50\}$.

Table 3. Experimental setup for the unified energy-aware benchmarking framework.

Results

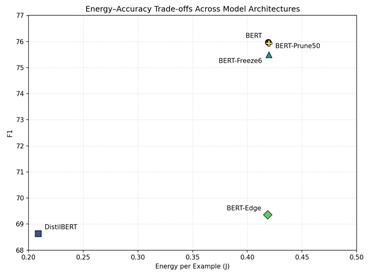

Figure 1. Inference energy per example vs. F1 across model architectures. Architectural variants largely preserve inference cost while trading off accuracy.

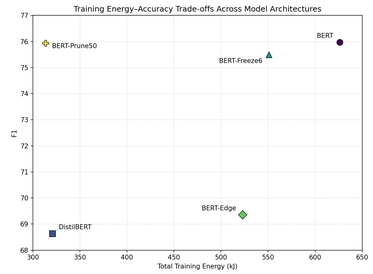

Figure 2. Training energy vs. F1 across model architectures. Architectural compression reduces training energy, with varying degrees of accuracy loss.

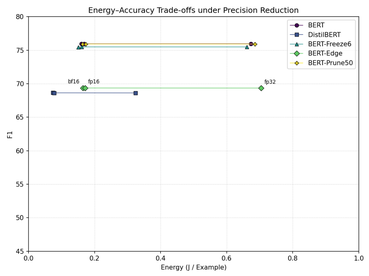

Figure 3. Precision reduction: inference energy per example vs. F1. Reduced precision dramatically lowers inference energy with minimal accuracy loss.

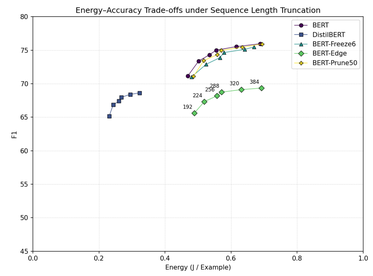

Figure 4. Sequence length truncation: inference energy per example vs. F1. Sequence length truncation produces a smooth energy–accuracy Pareto frontier.

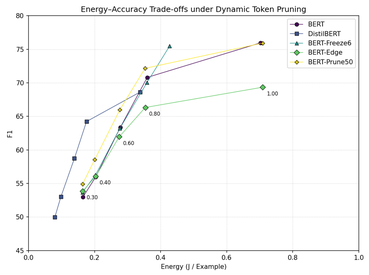

Figure 5. Dynamic token pruning: inference energy per example vs. F1. Token pruning yields large energy reductions but a steeper accuracy trade-off.

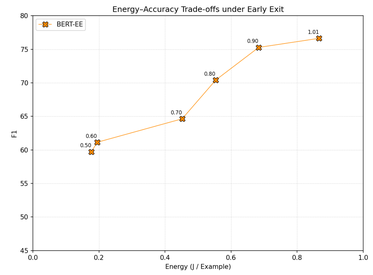

Figure 6. Early exit: inference energy per example vs. F1. Early exit adaptively reduces computation but requires careful threshold tuning.

Key findings:

- Architectural variants show limited impact on inference efficiency, but can substantially reduce training energy.
- Reduced precision provides the most favorable inference-time trade-off, lowering energy by up to $\approx 4\times$ with negligible accuracy loss.
- Sequence length truncation produces a smooth Pareto frontier, while token pruning and early exit introduce steeper energy–accuracy trade-offs.

Analysis

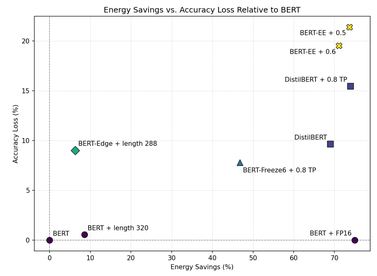

Figure 7. Energy savings vs. accuracy loss relative to the BERT baseline. Precision reduction dominates the efficiency frontier.

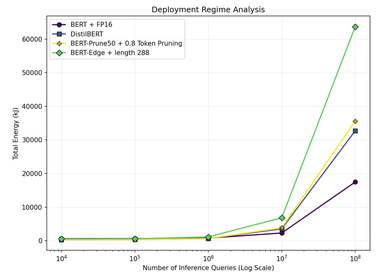

Figure 8. Total energy vs. number of inference queries across representative configurations. The optimal efficiency strategy depends on deployment scale.

Key takeaways:

- Different efficiency techniques operate on distinct parts of the transformer computation, explaining why they move models along the energy–accuracy frontier in qualitatively different ways.
- Techniques that reduce arithmetic cost (e.g., precision reduction) preserve model representations, whereas methods that remove input information or intermediate computation introduce stronger accuracy trade-offs.
- Selecting an efficiency strategy is ultimately a deployment decision: the most energy-efficient configuration depends on the expected inference workload and acceptable accuracy loss.

Conclusion

Contributions

- Energy–accuracy trade-offs in transformer models depend on both architectural design and inference-time configuration.
- Precision reduction emerges as the most effective inference-time optimization, dramatically lowering energy consumption with minimal impact on accuracy.
- The most efficient strategy ultimately depends on the deployment regime: training-efficient architectures are preferable for small workloads, while inference-efficient configurations dominate at large scale.

Limitations and future work

- Results are evaluated on BERT-family models and the SQuAD v2 task; extending the analysis to larger architectures and additional NLP tasks would test the generality of these findings.
- Future work could explore more aggressive precision formats (e.g., FP8 or 4-bit) and improved dynamic inference techniques such as advanced pruning or early-exit strategies.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova.
Bert: Pre-training of deep bidirectional transformers for language understanding.
In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

[2] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni.
Green ai.
Communications of the ACM, 63(12):54–63, 2020.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin.
Attention is all you need.
In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.