

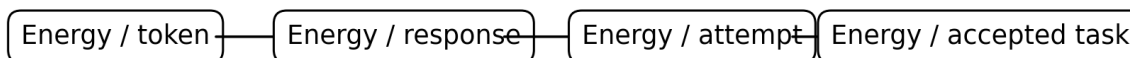
The Physical Cost of Useful AI Work

Research Brief | CL-WP-005 | v0.9 | September 2026

Critical Load asks a different question from most inference benchmarks: not “how many joules per token?”, but “how much measured energy is required to obtain an acceptable result?”

Why this matters

A user goal may trigger multiple model calls, long context, tools, retries and validation. Invocation-level metrics can therefore reward workflows that look efficient per call but consume more energy before producing an acceptable outcome.



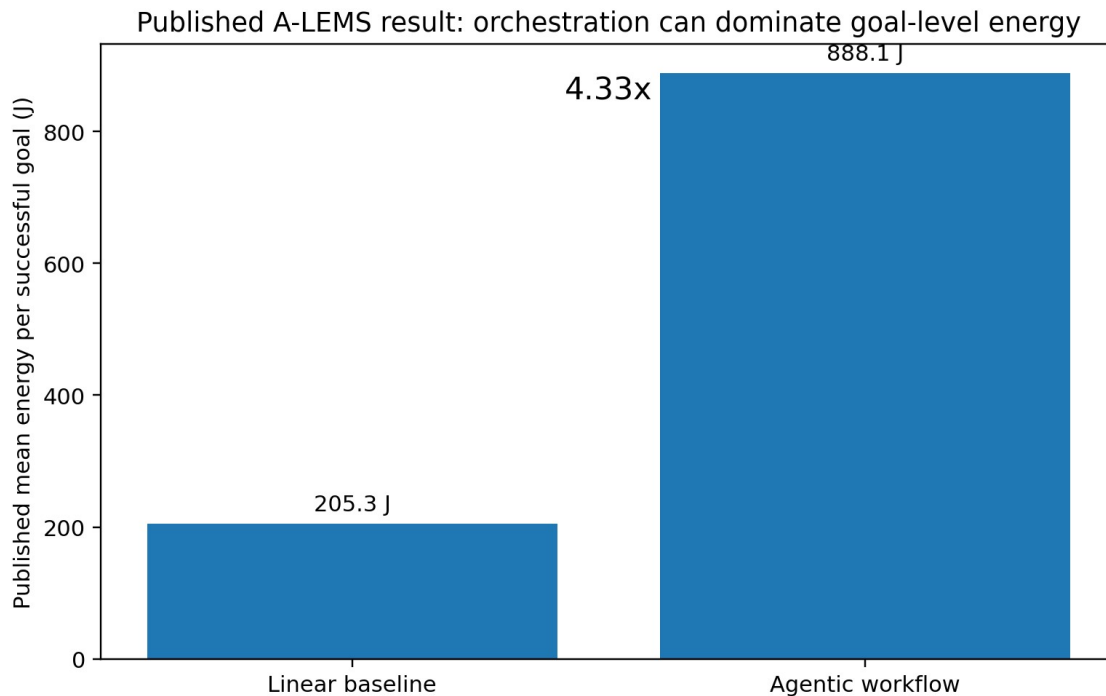
Critical Load thesis: normalize energy to the unit users actually value.

Proposed primary metric

Energy per Accepted Task = total measured energy across all attempts / accepted tasks. Report success rate, retry energy share, latency, tool calls and a quality-energy Pareto frontier alongside it.

What public evidence already shows

TokenPowerBench and ML.ENERGY show that inference energy can be measured reproducibly and varies by task, hardware and serving configuration. Agentic-energy research shows that goal-level orchestration changes the picture: A-LEMS reports 4.33x higher mean energy per successful goal for agentic workflows than matched linear baselines across its evaluated families, while web-agent research finds that higher energy does not necessarily mean better results. These are external results, not Critical Load measurements.



How to use the method

- Freeze the task and acceptance rule before comparing models.
- Measure energy at a declared boundary: GPU, CPU+GPU, node or facility.
- Include failures and retries in the numerator.
- Do not rank workflows that fail the quality gate as “efficient.”
- Use Pareto frontiers when quality, energy and latency trade off.

Release status

This v0.9 package is a methods and open-measurement synthesis release. It is ready to guide experiments and engineering requirements, but it is not yet a Critical Load-owned primary benchmark. v1.0 requires repeated measured runs, raw telemetry and independent review.