

The Physical Cost of Useful AI Work

Task-normalized energy accounting for multi-step and agentic AI workflows

CL-WP-005 | Release Candidate v0.9 | September 2026

Primary programme: R01 Workload Energy Attribution | Contributing programme: R03 AI Inference Intensity

STATUS

Release class	Methods + open-measurement synthesis working paper. Not a completed Critical Load primary benchmark.
Critical Load contribution	Defines a task-normalized accounting method, acceptance-gated benchmark protocol, measurement boundaries, release gates, and reproducible analysis tooling.
Primary measurement status	Not yet completed by Critical Load in this environment. Public measurement records are used only as external evidence and worked examples.
Use class	Benchmark design, measurement planning, model/workflow evaluation, energy-quality decision support, and procurement/engineering requirements.
Not suitable for	Claims about universal model efficiency, facility-level energy without facility measurement, or Critical Load-owned primary performance rankings before the measurement campaign is executed.

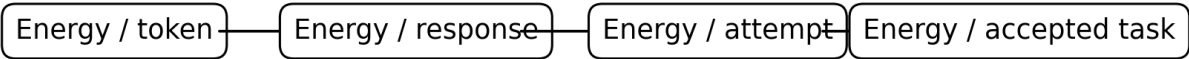
Executive summary

AI energy accounting is commonly normalized per token, request, or inference. Those units are useful for diagnosing serving efficiency, but they can misstate the physical cost of useful work when a user goal requires multiple calls, tool use, retries, long context, validation, and failure recovery. This paper defines a practical measurement system in which energy is attributed to an accepted task rather than to an invocation.

The central rule is simple: quality gates come before energy comparison. A workflow that uses fewer joules but fails the task is not more efficient at producing useful work. Critical Load therefore defines Energy per Accepted Task (EAT) as the primary comparison unit, with success rate, retry energy share, orchestration overhead, latency, and a quality-energy Pareto frontier reported alongside it.

The method is grounded in existing open measurement infrastructure. TokenPowerBench demonstrates GPU-, node-, and system-oriented measurement for LLM inference; ML.ENERGY provides a large public inference benchmark across 46 models, seven tasks, and 1,858 configurations; NVIDIA exposes cumulative GPU energy counters through NVML and telemetry through AIPerf. Recent agentic-energy research further shows why invocation-level normalization is insufficient: A-LEMS reports a 4.33x difference in mean energy per successful goal between agentic and matched linear baselines across its evaluated task families, while separate AAAI work on web agents finds that greater energy use does not necessarily imply better task performance.

This release does not claim that Critical Load has completed its own GPU benchmark. Instead, it establishes the measurement protocol, data model, reproducibility assets, task registry, validation gates, and publication standard required for a defensible primary campaign. That distinction is deliberate: a publishable methods result is preferable to fabricated precision.



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

Figure 1. The accounting unit moves from machine output toward user-valued work. Energy per accepted task is only meaningful when acceptance criteria are fixed before comparison.

Key findings and decisions

- **Token-normalized energy is diagnostic, not sufficient.** A token has no fixed user value: tasks differ in output length, correctness requirements, tool calls, and retry probability.
- **Success probability is an energy variable.** If a lower-energy attempt fails more often, total energy required to obtain one acceptable result can exceed that of a higher-energy but more reliable workflow.
- **Orchestration can add or save energy.** Agentic control is not intrinsically inefficient. It must be compared with a matched linear baseline under identical task criteria.
- **Measurement boundaries must remain explicit.** GPU-only energy cannot be relabelled as node or facility energy. Results should carry a B0-B3 boundary tag.
- **Decision outputs should be Pareto-based.** Quality, energy, and latency should remain visible rather than collapsed into one opaque composite score.
- **The next release gate is primary measurement.** A Critical Load-owned v1.0 benchmark requires repeated measured runs, frozen tasks, acceptance rules, and raw telemetry.

1. Research question

How much measured physical energy is required to complete a quality-accepted AI task, and how do model choice, reasoning, orchestration, retries, context length, and tool use change that cost?

The research is intentionally narrower than a full lifecycle assessment. It measures operational energy attributable to a declared compute boundary during a defined workflow. Carbon, water, and location effects can be added later through separate Critical Load layers once the physical energy record is stable.

2. Why existing normalization is incomplete

Energy per token is valuable for inference-engine tuning because it isolates a machine-level output unit. Energy per response is better for a request-level cost view. Neither captures whether the response solved the task, whether another response was required, whether a tool call failed, or whether a validator forced a retry.

A practical benchmark therefore needs two layers: measurement of physical energy at the hardware boundary, and evaluation of task acceptance at the application boundary. The join between those layers is the core methodological problem addressed here.

3. Evidence landscape

3.1 Open inference measurement

TokenPowerBench, published at AAAI 2026, provides a lightweight benchmark that can measure GPU-, node-, and system-level power and attributes energy to prefill and decode phases. Its public repository includes measurement results and power-monitoring code. The Critical Load protocol does not replace this layer; it extends the normalization boundary from inference to accepted work.

ML.ENERGY v3 reports 46 models across seven tasks and 1,858 configurations on NVIDIA H100 and B200 hardware. This scale demonstrates that energy varies materially with model, task, hardware, and configuration, and that no single joules/token value should be treated as a universal property of a model.

3.2 Agentic and goal-level evidence

A-LEMS explicitly proposes Energy per Successful Goal for agentic systems. The preprint reports 888.1 J for agentic workflows versus 205.3 J for linear baselines on average across its evaluated families, a 4.33x difference, while tool-augmented tasks can reverse the direction of the orchestration overhead index. The appropriate inference is not that agents always waste energy, but that orchestration structure must be measured at goal level.

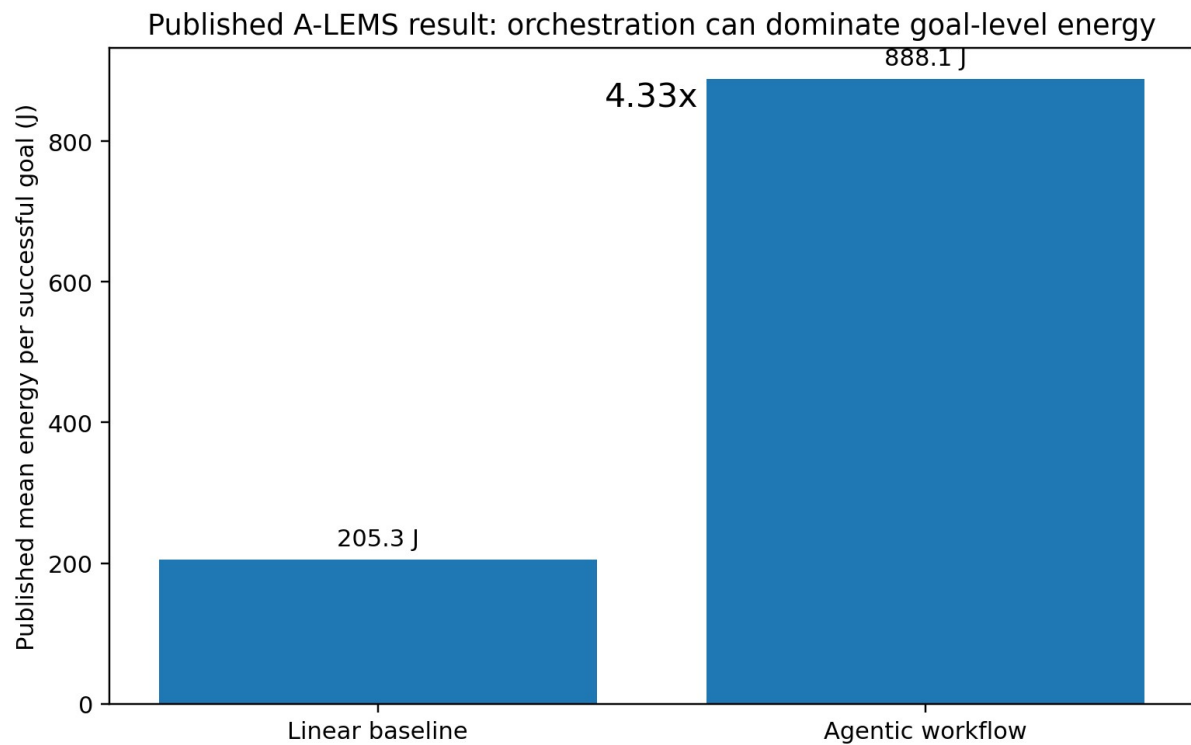


Figure 2. Published A-LEMS summary result. Values are external evidence, not Critical Load measurements.

Krupp et al. independently benchmark web agents and report that higher energy consumption does not necessarily correspond to better results. That observation motivates a quality-gated rather than energy-only comparison.

4. Worked open-measurement example

One public TokenPowerBench result for Llama-3.1-8B under vLLM, batch size 128 and a reported 500-token target records 432.76 W mean GPU power, 1,675.26 J GPU energy, 11,423 output tokens across 32 responses, 0.1467 J per output token and 111.68 J per response. CPU and DRAM readings in that record are zero, so the correct boundary is GPU only. Critical Load does not reinterpret the zero CPU/DRAM values as proof of zero non-GPU energy.

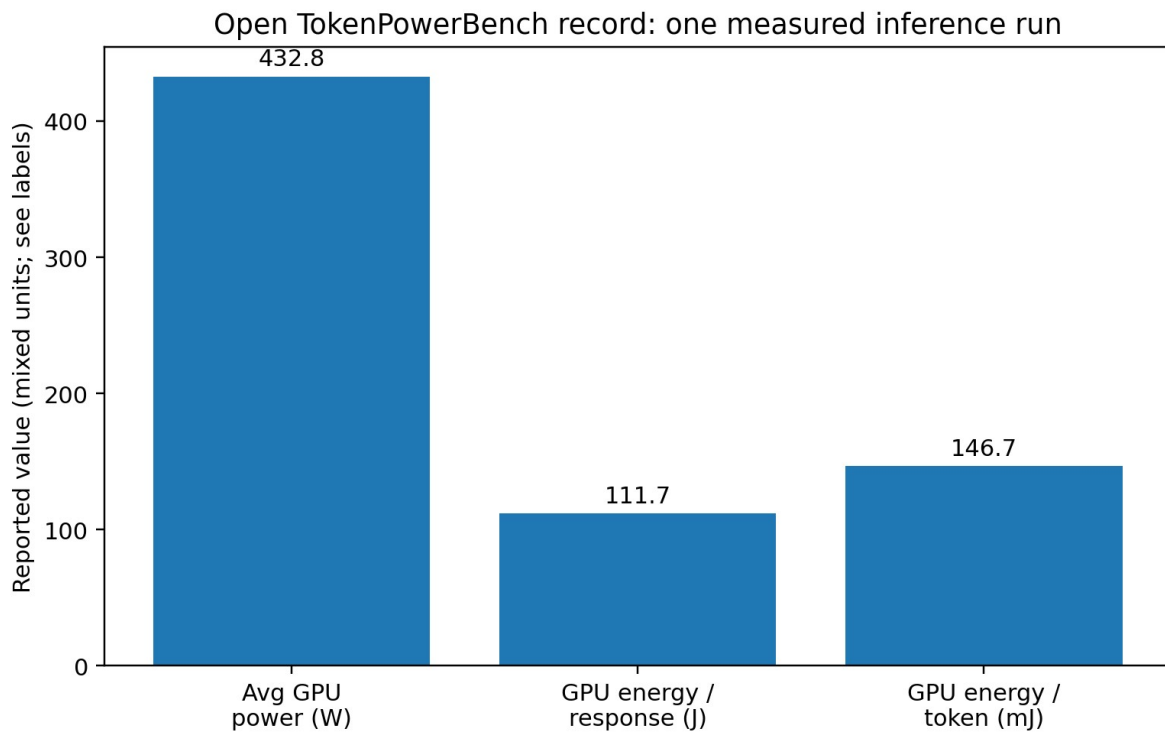


Figure 3. External open measurement used as a worked example. Mixed units are shown only to demonstrate the distinction between power, per-response energy and per-token energy.

5. Critical Load Useful AI Work Measurement (UAWM) method

5.1 Unit of analysis

A task is the smallest user-valued objective with a frozen input, acceptance rule, and maximum-attempt policy. An attempt is one execution of a workflow toward that task. A task is accepted only when the predeclared scoring rule is satisfied. Tool calls and model invocations are implementation events inside an attempt, not the reporting unit.

5.2 Primary metric

Energy per Accepted Task (EAT) = total measured energy across all attempts / number of accepted tasks. The numerator includes failed attempts and retries within the declared benchmark boundary. The denominator excludes unaccepted tasks rather than pretending they delivered useful output.

5.3 Companion metrics

Attempt success rate	Accepted attempts / total attempts. Report with uncertainty.
Retry energy share	Energy spent after first attempts / total workflow energy.
Orchestration overhead index	Energy per accepted task for agentic workflow / matched linear baseline.
Energy per token / response	Diagnostic metrics only; never substitute for

	accepted-task energy.
Quality-energy frontier	Pareto set across acceptance/quality and energy; preferred decision display.
Latency	Wall-clock time per attempt and per accepted task.
Tool-call count	Number and type of external actions used by the workflow.

6. Measurement boundaries



Figure 4. Boundary labels used by Critical Load. A wider boundary may subsume a narrower one only when the wider energy is actually measured or defensibly allocated.

B0 - GPU only: Use cumulative GPU energy counters (preferred) or time-integrated power. Suitable for accelerator-attributable comparisons on the same platform.

B1 - CPU + GPU: Add CPU package and memory/DRAM energy where RAPL or equivalent is supported. Keep each component visible.

B2 - Node: Use PDU/BMC/full-node metering where available. This is the preferred operational comparison boundary for deployment decisions.

B3 - Facility: Allocate measured facility energy using declared metering and allocation rules. Do not create B3 results by multiplying B0 by an assumed PUE unless the output is explicitly labelled modelled rather than measured.

7. Experimental design for the primary campaign

The release campaign should use at least four task families and should freeze inputs, expected outputs, scoring rules and maximum attempts before collecting comparative energy data. Suggested families are structured extraction, research synthesis, coding with executable tests, long-context question answering, deterministic tool workflows, and a synthetic/public compliance workflow representative of multi-step professional work.

Family	Acceptance	Energy unit	Minimum repeats	Important confounder	Primary use
Extraction	Schema + field F1 threshold	J/accepted document	≥ 3 independent runs	Document length	Operational automation
Synthesis	Claim/citation rubric	J/accepted synthesis	≥ 3	Source count/context	Research workflows
Coding	All tests pass	J/accepted solution	≥ 5	Sandbox/runtime	Developer tools
Long context	Dataset score threshold	J/accepted answer	≥ 3	Context length	Enterprise document AI
Tool workflow	Target final state	J/accepted goal	≥ 5	Tool latency/failures	Agentic systems
Compliance workflow	Fixed checklist	J/accepted deliverable	≥ 3	Evidence volume	Professional multi-step work

8. Run protocol

1. Lock hardware, driver, CUDA/runtime, inference engine, model revision, precision, prompt template, temperature, seed policy and concurrency.
2. Warm the model and exclude warm-up energy from the profiling window.
3. Record cumulative energy counters immediately before and after each attempt; collect sampled power for diagnostic validation.
4. Log every model call, tool call, retry, validator decision and wall-clock timestamp.
5. Score the final artifact using the frozen acceptance rule without reference to its energy use.
6. Repeat each task/model/workflow combination on at least three independent benchmark runs; use more repeats when task stochasticity is material.
7. Report raw values, aggregate statistics, confidence intervals, exclusions and all failed attempts.
8. Publish the measurement boundary next to every energy number.

9. Statistical and quality requirements

Primary comparisons should not be based on a single run. For each task-model-workflow cell, report the number of independent benchmark runs, attempt count, acceptance rate, mean/median energy per accepted task, dispersion, and confidence interval. Outlier removal, if any, must be rule-based and disclosed before model comparison.

Quality should not be compressed into a single energy-efficiency ratio unless a use case explicitly warrants it. The default presentation is a Pareto frontier: models/workflows dominated on both quality and energy can be rejected; trade-offs among non-dominated alternatives remain visible to decision makers.

10. Practical decision uses

Model routing: Choose the lowest-energy workflow that still clears the acceptance threshold for a task class.

Reasoning policy: Enable longer reasoning only where its quality gain moves the workflow onto a better quality-energy frontier.

Agent design: Measure whether planning, reflection, parallel branches and tool retries save or consume net energy at goal level.

Procurement: Require vendors to report measured node energy per accepted workload rather than only peak GPU TDP.

Capacity planning: Convert accepted-task demand forecasts into measured operational energy before adding location-specific grid, water and carbon layers.

Engineering regression: Treat energy per accepted task as a release metric alongside latency and success rate after workflow changes.

11. What this release establishes - and what it does not

Established

- A formal task-normalized energy accounting framework.
- A measurement boundary taxonomy.
- A reproducible run-log schema and metric computation script.
- A primary benchmark task registry template.
- Release validation rules that reject under-replicated or boundary-ambiguous data.
- A source register separating external evidence from Critical Load contribution.

Not yet established

- Critical Load-owned GPU or node telemetry across competing models.
- A model ranking or recommendation.
- A universal energy factor for agents or reasoning.
- Facility-level energy per accepted task.
- Cross-hardware generalization.

12. Release gate for CL-WP-005 v1.0

G1 Task lock	At least four public task families with frozen inputs and acceptance criteria.
G2 Boundary lock	GPU-only or node measurement selected and hardware/software manifest published.
G3 Measurement lock	Raw energy telemetry captured for every attempt, including failures.
G4 Replication	At least three independent runs per core comparison cell; more for stochastic agent workflows.
G5 Scoring audit	Automated scoring validated; a sample of complex-

	task decisions independently reviewed.
G6 Reproduction	A clean environment reproduces summary tables and figures from raw logs.
G7 External review	At least one independent systems/energy reviewer checks measurement and one task-domain reviewer checks acceptance rules.

13. Recommended v1.0 benchmark matrix

The first primary campaign should be deliberately narrow: two or three open models, one hardware family, one inference engine, and four task families. The objective is to establish the measurement method, not to produce a sprawling leaderboard. A useful initial design is 3 models x 4 tasks x 2 workflow modes x 5 repeated task instances x 3 independent runs. At three attempts maximum, this remains tractable while generating enough failure and retry information to test task-normalized accounting.

14. Reproducibility assets in this package

- data/public_evidence_register.csv
- data/tokenpowerbench_reference_record.csv
- data/task_registry_template.csv
- data/run_log_template.csv
- data/metric_definitions.csv
- code/measure_nvml.py
- code/compute_metrics.py
- code/validate_release.py
- CL_UAW_Benchmark_Protocol.xlsx
- CITATION.cff
- METHOD.md
- RELEASE_REVIEW.md

15. References

- [1] Niu, C. et al. (2026). TokenPowerBench: Benchmarking the Power Consumption of LLM Inference. AAAI-26. DOI: 10.1609/aaai.v40i38.40535.
- [2] Chung, J.-W. et al. (2025). The ML.ENERGY Benchmark: Toward Automated Inference Energy Measurement and Optimization. NeurIPS Datasets and Benchmarks; leaderboard v3.0 update January 2026.
- [3] Panigrahy, D. & Tyagi, A. (2026). Energy per Successful Goal: Goal-Level Energy Accounting for Agentic AI Systems. arXiv:2605.22883.
- [4] Krupp, L. et al. (2026). Promoting Sustainable Web Agents: Benchmarking and Estimating Energy Consumption Through Empirical and Theoretical Analysis. AAAI-26. DOI: 10.1609/aaai.v40i45.41221.
- [5] NVIDIA. NVML API Reference: nvmlDeviceGetTotalEnergyConsumption. Accessed September 2026.
- [6] NVIDIA. AIPerf GPU Telemetry and Power Efficiency Documentation. Accessed September 2026.

[7] Chenxu Niu et al. TokenPowerBench public GitHub result: Llama-3.1-8B vLLM batch 128 / 500-target record. Accessed September 2026.

Citation

Critical Load Research Group (2026). The Physical Cost of Useful AI Work: Task-normalized energy accounting for multi-step and agentic AI workflows. CL-WP-005, Release Candidate v0.9. Critical Load. September 2026.