INSTITUTE FOR PHYSICAL AI @ BMI
The Charlot Lab

Technical Report TR-2026-06
Survey / Review · Preprint v1
7 July 2026

MATHGROUND

MathGround: Pricing Every Decision in Joules, with a Replayability Class on Every Answer

A substrate that resolves each request at the lowest tier whose grammar covers it, meters the cost in measured energy, and stamps every claim with a type-enforced replayability class.

David Jean Charlot, PhD

Dean of Physical AI · The Charlot Lab, Institute for Physical AI @ BMI

Correspondence: contact@physicalai-bmi.org · physicalai-bmi.org
Interactive companion: physicalai-bmi.org/research/charlot-lab#topic-mathground

Abstract. The cost of intelligence is energy, and most of what a system must decide is not a generation problem. This report describes MathGround, the substrate the remainder of the lab's work runs on. Two design commitments organize it. First, every decision is priced in joules and a request resolves at the lowest tier whose grammar covers it: a deterministic lookup, then a closed-form formula, then a sparse solver, and a stochastic model only as a last resort. On commodity silicon the model tier costs orders of magnitude more energy than a cited composition, so declining to invoke it unless the request demands it is where the joules are saved. Second, every claim carries a replayability class — deterministic, retrieved-and-cited, composed, or model-generated — enforced at the type level, so a generated answer can never be relabeled as ground truth. Each answer returns with a receipt: a measured energy figure and a class, signed on-device with an elliptic-curve key. The same substrate carries the lab's robot-policy harness, placing every frontier architecture behind one action space and one energy meter. This is a research position and a system description; it reports no original experiments.

1. Introduction

The dominant cost of contemporary artificial intelligence is energy. At the device level the expense is arithmetic and, more than arithmetic, the movement of operands, where a memory fetch can exceed a floating-point operation by orders of magnitude[1]. At the system level the training and serving of large models carries an energy and carbon cost large enough to be a first-order design concern[6,7]. The default posture of the current tooling is to route almost

every request through the most expensive mechanism available, a large generative model, regardless of whether the request needs it.

Most of what an embodied or analytical system must decide is not a generation problem. Converting a unit, evaluating a known constant, integrating a rate over an interval, checking a bound, or solving a small linear system are tasks with exact, cheap answers. A generative model can produce those answers, but it produces them by sampling, at high energy cost, and without a guarantee that the answer is the exact one rather than a plausible one. The mismatch is not a matter of accuracy alone. It is a matter of the unit of account: when the cost of a decision is invisible, there is no pressure to resolve it cheaply, and no record of how it was resolved.

MathGround takes energy as the unit of account and makes two commitments. A request resolves at the lowest tier whose grammar covers it, and every answer it returns is stamped with the class of process that produced it. The reference implementation, at `mathground.ai`, prices energy against real device power measured on the silicon it runs on and signs each receipt with an on-device key. This report describes the substrate, its resolution cascade, its type-enforced provenance, and the robot-policy harness built on the same meter.

2. Scope and method

This is a review and a research position accompanied by a system description. It states the design of an operating substrate and the reasoning behind it; it does not report original experiments, benchmark numbers, or measured results. Where the report gives energy figures for the tiers of the cascade, they are illustrative orders of magnitude drawn from the published per-operation and data-movement costs of commodity silicon[1], not measurements of the system, and they are marked as such. The reference implementation measures its own energy at run time against device power, but those measurements are a property of the running product and are not tabulated here as findings. The report cites primary sources for each external claim. Its principal limitation is that the substrate is a prototype in active development, and several of the mechanisms described, in particular the robot-policy harness, are early-stage rather than settled results.

3. The tiered resolution cascade

The organizing structure of MathGround is a cascade of four resolution tiers, ordered by cost. A request is presented to each tier in turn, from cheapest to most expensive, and is resolved by the first tier whose grammar covers it. The tiers are: a deterministic lookup, over constants, units, tables, and previously cached exact results; a closed-form formula, when the request matches a known analytic expression; a sparse numerical solver, for small linear systems, root-finding, and quadrature; and a stochastic model, invoked only when no lower tier's grammar applies. The principle is not new in kind. Model cascades and learned routers have been shown to cut the cost of language-model serving by sending easy queries to cheap responders and reserving the expensive model for the hard remainder[4,5]. MathGround differs in three ways: the cheap tiers are exact rather than smaller approximate models, the routing gate is a

grammar-coverage test rather than a learned confidence score, and the axis of optimization is measured joules rather than API price.

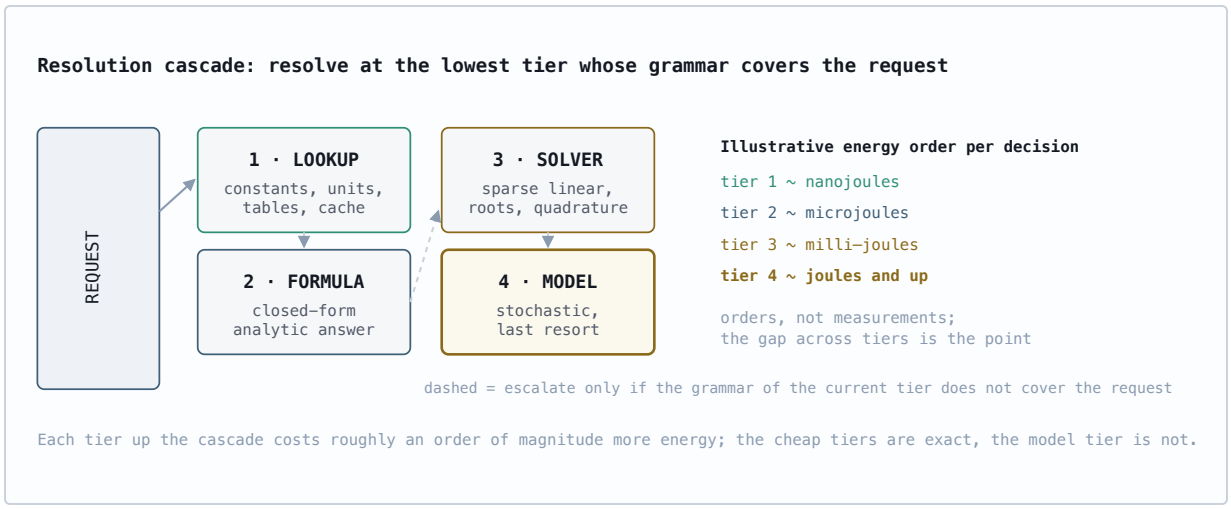

Figure 1. The four-tier resolution cascade. A request is resolved by the lowest tier whose grammar covers it and escalates only when that grammar fails to apply. The energy figures are illustrative orders of magnitude consistent with published per-operation and data-movement costs on commodity silicon[1], not measurements of the system. An interactive companion is at physicalai-bmi.org/research/charlot-lab.

The energy argument is the reason for the ordering. A lookup or a closed-form evaluation resolves in arithmetic and a small number of memory accesses; a generative pass through a large model resolves in billions of multiply-accumulate operations and their attendant data movement, whose per-operation cost is the subject of a well-known account of computing's energy problem[1]. The separation between the tiers is not marginal. It is several orders of magnitude, which is why the discipline of not invoking the model tier is where the joules are actually saved. A cascade that occasionally escalates a hard request to the model, but resolves the common case in the exact tiers, spends its energy budget where it must and nowhere else.

The expected energy of a request under the cascade is a weighted sum over the tiers, where each tier's weight is the probability that the request first resolves there:

$$\mathbb{E}[E] \;=\; \sum_{t=1}^{4} p_t \, E_t, \qquad E_1 \ll E_2 \ll E_3 \ll E_4, \qquad \sum_{t} p_t = 1.$$

Because E_4 dominates the E_t by orders of magnitude, the expected cost is governed almost entirely by p_4, the fraction of requests that reach the model tier. Reducing that fraction, rather than making the model cheaper, is the lever with the most leverage. The grammar-coverage gate exists to keep p_4 small without sacrificing correctness on the requests the lower tiers can answer exactly.

4. The replayability class as a type

An energy figure answers how much a decision cost. It does not answer what kind of thing the answer is. MathGround attaches to every claim a second attribute, its *replayability class*, which

records the process that produced it and how, if at all, it can be reproduced. There are four classes. A **deterministic** claim is one a lookup or a pure function produced, reproducible bit-for-bit by re-running the same computation. A **retrieved-and-cited** claim is a value drawn from a named source, reproducible by consulting that source. A **composed** claim is one assembled from lower claims by an exact operation, reproducible by replaying the composition over its inputs. A **model-generated** claim is a sample from a stochastic model, reproducible only in distribution and only given the same seed and weights.

The classes are ordered by the strength of the guarantee they carry, and the ordering is the point. The critical property is that a model-generated claim can never be relabeled as ground truth. MathGround enforces this at the type level: the class is part of the type of a claim, and the only operations that produce a deterministic or retrieved-and-cited claim are the deterministic and retrieval tiers themselves. A composition over inputs takes the weakest class among its inputs, so a composition that touches a model-generated value is itself model-generated and cannot silently become composed or deterministic. There is no cast that launders a lower class into a higher one. This is the same discipline a type system applies to units or to tainted input: the property travels with the value and the compiler refuses the programs that would violate it.

Table 1. The four replayability classes, the tier that mints each, and the reproduction guarantee it carries.

CLASS	MINTED BY	REPRODUCTION GUARANTEE
Deterministic	Lookup / formula (tiers 1–2)	Bit-for-bit on re-run of the same computation.
Retrieved-and-cited	Retrieval over a named source	Reproducible by consulting the cited source.
Composed	Exact composition (tier 3)	Reproducible by replaying the composition; takes the weakest class of its inputs.
Model-generated	Stochastic model (tier 4)	Reproducible only in distribution, given the same seed and weights.

The value of the class is most visible when it is weak. A model-generated number that looks like a measured constant is exactly the failure mode that erodes trust in machine reasoning. Carrying the class with the value, and refusing to relabel it, means the caller always knows whether an answer is ground truth, a citation, a derivation, or a guess.

5. The joules receipt and on-device attestation

Every answer returns with a receipt. The receipt records the energy the request consumed, the tiers it traversed, the replayability class of the result, and a cryptographic signature over those fields. In the reference implementation the energy is not estimated from a model of the hardware but measured against real device power sampled from the silicon at run time, so the figure on the receipt is a property of the actual computation rather than a nominal cost.

The receipt is signed on-device with an elliptic-curve digital signature[8]. Signing on the device that performed the computation, with a private key that does not leave it, binds the energy figure and the class to the machine that produced them: a receipt can be checked by anyone holding the public key, and it cannot be forged by a party that did not do the work. This is attestation of a claim about cost and provenance rather than attestation of a program's identity, but it uses the same primitive and gives the same non-repudiation. The signature is what turns a self-reported energy number into evidence.

> The physical floor under the energy ledger is Landauer's bound, the minimum `kT ln 2` of heat generated by erasing one bit of information[2]. Commodity silicon operates many orders of magnitude above this bound, so the joules a receipt reports are set by architecture and data movement, not by thermodynamics. The bound matters here as the reference point that makes the reported figures interpretable: it is the reason a cheaper tier is cheaper for a physical reason, not merely a bookkeeping one.

6. The robot-policy harness

The same substrate carries the lab's robot-policy harness. A control policy for an embodied system is, in the terms of this report, a request that resolves at some tier: a lookup into a cached trajectory, a closed-form controller, a solver step, or a sample from a learned generative policy. Placing policies on MathGround means placing every candidate architecture behind one action space and one energy meter. A frontier world-action model, of the kind exemplified by world foundation models built for Physical AI[3], sits at the model tier of the cascade; a scripted or analytic controller sits at a lower tier; and both are measured by the same meter and stamped with the same class.

The harness makes two questions answerable that are otherwise confounded. The first is comparative energy: what does a given architecture cost per decision, on the same silicon, under the same action space, relative to a cheaper baseline that resolves the same request. The second is provenance: which of a policy's decisions were exact and which were sampled, so that a controller's guarantees are legible rather than assumed. Both questions reduce to the receipt. This part of the substrate is early-stage; the report describes the harness as a design, not as a demonstrated benchmark, and reports no policy measurements.

7. Position and discussion

The position of this report is narrow and can be stated plainly. The energy problem of intelligence is well described at the device level[1] and at the system level[6,7], and the response of cascade and routing work is to send cheap requests to cheap responders[4,5]. That work leaves an axis open. It optimizes an approximate objective, usually API price or a learned confidence, and it routes among models that are all generative and therefore all of the same replayability class. MathGround takes the open axis: energy as the unit, measured rather than estimated; a resolution gate based on exact grammar coverage rather than a confidence score; a type-enforced replayability class so that provenance is not a convention but a property the

compiler enforces; and an on-device signed receipt so that both the cost and the class are evidence.

The design has costs of its own. Grammar coverage is only as broad as the grammars written, and a request the lower tiers cannot parse escalates to the model tier whether or not a cheaper exact answer exists in principle. The four classes are a coarse partition of a richer space of provenance. Measured energy is faithful to the device it runs on and therefore not portable across devices without care. None of these undermines the thesis, which is a claim about where the joules are and how to make cost and provenance legible, not a claim that the substrate is complete. The claim is that pricing decisions in joules, resolving them at the lowest adequate tier, and typing every answer with its replayability class are the right invariants for the rest of the lab's work to stand on.

8. Conclusion

MathGround is the substrate the remainder of the lab's work runs on. It prices every decision in joules, resolves each request at the lowest tier whose grammar covers it, and returns an on-device signed receipt carrying a measured energy figure and a type-enforced replayability class. Because the model tier costs orders of magnitude more energy than a cited composition on commodity silicon, the discipline of not invoking it is where the energy is saved, and because a generated answer cannot be relabeled as ground truth, the caller always knows what kind of answer it holds. The robot-policy harness places every frontier architecture behind one action space and one energy meter on the same substrate. The report states a design and a position; it reports no original measurements, and it marks the harness and the wider substrate as prototypes rather than settled results.

References

1. M. Horowitz. 1.1 Computing's energy problem (and what we can do about it). IEEE International Solid-State Circuits Conference (ISSCC), 2014. doi:10.1109/ISSCC.2014.6757323.
2. R. Landauer. Irreversibility and heat generation in the computing process. IBM Journal of Research and Development, 5(3):183–191, 1961. doi:10.1147/rd.53.0183.
3. NVIDIA. Cosmos World Foundation Model Platform for Physical AI. arXiv:2501.03575, 2025.
4. L. Chen, M. Zaharia, J. Zou. FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. arXiv:2305.05176, 2023.
5. I. Ong, A. Almahairi, et al. RouteLLM: Learning to Route LLMs with Preference Data. arXiv:2406.18665, 2024.
6. E. Strubell, A. Ganesh, A. McCallum. Energy and Policy Considerations for Deep Learning in NLP. arXiv:1906.02243, 2019.
7. D. Patterson, J. Gonzalez, et al. Carbon Emissions and Large Neural Network Training. arXiv:2104.10350, 2021.
8. National Institute of Standards and Technology. Digital Signature Standard (DSS). FIPS PUB 186-5, 2023. doi:10.6028/NIST.FIPS.186-5.

AI-use disclosure. Preparation of this report used a large language model (Claude, Anthropic) for drafting and editing text, organizing the reviewed literature, and preparing the figures and the interactive companion. Cited

references were checked to resolve to their sources. The author reviewed the content and is solely responsible for it. Consistent with ICMJE, COPE, and IEEE guidance, the model is a tool and is not credited as an author.

Institute for Physical AI @ BMI · The Charlot Lab
503 McKeever Rd, Arcola, TX 77583, USA
physicalai-bmi.org · contact@physicalai-bmi.org

Technical Report TR-2026-06 · Preprint v1