

THE COMPUTABLE WORLD MODEL

The Computable World Model: Retrieving the Physics an Embodied Agent Needs

Most of the physics an embodied agent needs is retrievable, not computable. A single cascade, over a single graph, from a milliwatt chip to a workstation.

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-cad

ABSTRACT. A world model is only actionable when the system can answer physical questions about it: will this part hold, will it fit, will it overheat, how far will it deflect. Full computer-aided design and physics simulation can answer such questions, but their kernels and solvers are too heavy to run inside a body's power budget. This report states and develops a research position, that of the CadFuture project of the Charlot Lab: the physics an embodied agent needs at run time is mostly retrievable rather than computable. A query walks one ordered cascade, a stored lookup first, then a closed-form engineering formula, then a sparse local solver, and only as a last resort a learned surrogate or a full model. The cascade runs over one graph that carries geometry, the physics fields, tolerances, live digital-twin state, and the agent's own interaction history, so that each tier can reuse what the previous ones deposited. The same engineering model is intended to run from a sub-five-milliwatt neuromorphic chip to a workstation: execution adapts to the available energy, the engineering truth does not. The report reviews the primary sources this position draws on, gives an expected-cost argument for the cascade and a standard error bound for its surrogate tier, and is explicit about what is prototype and what is conjecture. No proprietary or novel experimental results are reported.

1. Introduction

Recent work on world models has concentrated on prediction in perceptual space: given what an agent has seen and done, predict what it will see next. For a body that must act, this is necessary but not sufficient. Acting on the physical world requires answers to physical questions about it. Will this bracket carry the load. Will the gripper clear the fixture. Will the

motor overheat at this duty cycle. Will the shelf deflect past its tolerance. These are not questions about pixels. They are questions about stress, contact, heat, and geometry, and a world model that cannot answer them is a model of appearances rather than of consequences.

The engineering disciplines already answer these questions, with computer-aided design and physics simulation. A boundary-representation kernel holds exact geometry and topology^[1]; the finite element method turns a partial differential equation over that geometry into a large sparse linear system and solves it^[2,3]. The difficulty for Physical AI is not accuracy but cost and location. A meshing and assembly step, a nonlinear contact solve, or a transient thermal run is measured in seconds to hours on a workstation, and in joules to kilojoules of energy. An embodied agent needs an answer in milliseconds, many times per second, inside a power envelope that for the smallest platforms is a few milliwatts. The full computation cannot be where the body is.

The usual response is to move the computation off the body, to a server, or to replace it with a single learned network trained to imitate the solver. Both have real uses and both are surveyed below. Neither, on its own, gives an embodied agent a physical world model that is at once fast, local, auditable, and faithful to the same engineering definitions a human engineer would use. This report develops a third position.

2. Scope and method

This is a review and a research position, not an experimental report. It surveys established primary sources on solid modeling, the finite element method, model order reduction, neural operators, and the design and analysis of computer experiments, and it situates against them the design claim of CadFuture, a Charlot Lab project (repository at github.com/dcharlot-physicalai-bmi/cad-future). The mathematical statements in Sections 4 and 6 are standard results restated for this setting, attributed to their sources, and no benchmark numbers are reported as findings. The energy figures in Figure 1 are order-of-magnitude illustrations of why the tiers are ordered as they are, not measurements of a specific implementation. Where CadFuture is a prototype or the claim is a conjecture rather than a demonstrated result, the text says so. The position composes with two other Charlot Lab lines, MathGround, which supplies the tiered evaluation cascade, and Graph of the World, which supplies the single graph; those are named where they carry the argument and are not re-derived here.

3. Retrievable versus computable physics

The central claim is a claim about the distribution of queries, not about the limits of computation. Everything an agent might ask is in principle computable from first principles. But the questions a working body actually asks are heavily concentrated. They repeat across time as the agent revisits the same joint, the same part, the same contact. They repeat across space as parametric families, the same bracket at a slightly different length, the same bearing at a slightly higher load. And they are usually decision questions with a coarse tolerance, will this hold with margin, not what is the stress field to four significant figures. A query drawn

from a concentrated, parametric, coarse-tolerance distribution is usually answerable by retrieving and adjusting a previous result, and only rarely by solving the governing equations afresh.

Retrieval here is meant in a strong sense. It includes an exact stored result for a query already seen; a closed-form engineering relation, such as a beam-deflection or a thermal-resistance formula, that returns an answer in a handful of operations; and a reduced or surrogate model that reconstructs a field from a small precomputed basis. Engineering practice has always leaned this way. Handbooks of standard cases, design charts, and allowable-stress tables are precomputed physics, and the statistical design of computer experiments formalized the idea of replacing an expensive simulator with a cheap interpolating surrogate fitted to a designed set of runs^[4]. CadFuture's position is that an embodied agent should treat the full solver the way an experienced engineer treats it, as the tool of last resort behind a handbook, a formula sheet, and memory, rather than as the default.

The claim is falsifiable. If, for a given embodiment and task, the query stream is not concentrated, if most queries are novel, high-precision, and non-parametric, then retrieval buys little and the cascade degenerates to running the solver every time. The position is therefore an empirical bet about embodied workloads, and it should be evaluated as one.

4. The one cascade: lookup, formula, sparse solver, model

Every physical query walks the same ordered cascade, supplied by MathGround. It is tried against the cheapest admissible method first and escalates only when the current tier cannot answer within the requested tolerance. Tier 1 is a lookup, an exact or nearest stored result carrying its own validity conditions. Tier 2 is a closed-form engineering formula, evaluated in constant time. Tier 3 is a sparse local solver, a finite element or finite difference solve restricted to the affected region rather than the whole assembly^[3]. Tier 4, the last resort, is a full model, either a reduced-order or neural surrogate^[5,6,7] or, where nothing cheaper is admissible, the full high-fidelity solve. Each tier that runs deposits its result back into the graph of Section 5, so tomorrow's identical query resolves at Tier 1.

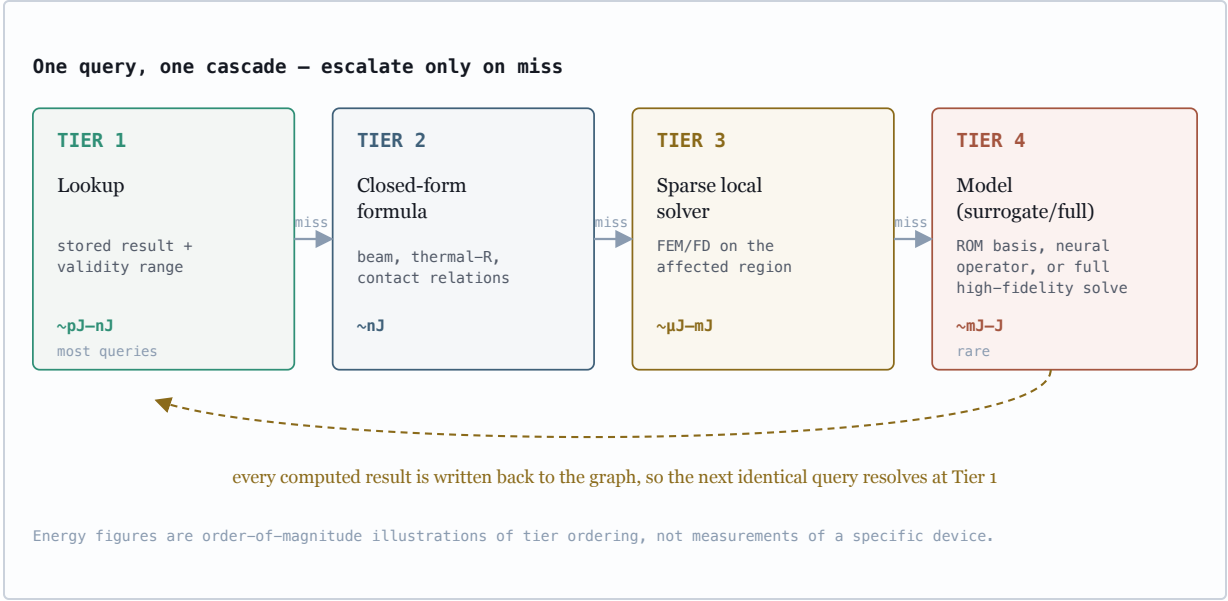


Figure 1. The four-tier retrieval cascade. Cost rises by roughly three orders of magnitude per escalation, so the expected cost is dominated by the cheapest tier that most queries reach. Results deposited back into the graph turn a Tier-3 or Tier-4 computation today into a Tier-1 lookup tomorrow. An interactive version is at physicalai-bmi.org/research/charlot-lab.

The ordering is justified by an expected-cost argument. Let the tiers have per-answer costs $c_1 < c_2 < c_3 < c_4$, and let q_k be the probability that a query is first answered at tier k , with $\sum_k q_k = 1$. Because a miss at a tier still pays that tier's probe cost before escalating, the expected cost of a query is

$$\mathbb{E}[C] = \sum_{k=1}^4 q_k \left(\sum_{j=1}^k c_j \right) \leq c_1 + \sum_{k=2}^4 \left(\sum_{j \leq k} q_j \right) c_k.$$

When the query distribution is concentrated, most of the mass sits on small k , so $\mathbb{E}[C] \approx c_1$ even though c_4 may be a million times larger. The cascade is worth building precisely to the extent that Section 3's concentration claim holds; if it fails and $q_4 \rightarrow 1$, the bound collapses to running the full model plus the wasted probe cost of the cheaper tiers. The design bet is that probe costs at Tiers 1 and 2 are small enough that this overhead is negligible when they usually hit.

5. One graph: geometry, fields, tolerances, twin state, interaction

The cascade only pays off if each tier can cheaply reuse what the others produced, and that requires a shared representation. CadFuture, drawing on the Graph of the World line, holds everything in a single typed graph rather than in separate CAD, simulation, and log files. Nodes and edges carry the boundary-representation geometry and topology^[1]; the physics fields attached to that geometry, whether measured, computed, or interpolated; the tolerances and allowables that turn a field into a pass-or-fail decision; the live state of the physical digital twin, the as-built and as-sensed deviations from nominal^[8]; and the agent's own interaction

history, which queries it asked, which contacts it made, which results were deposited. Placing these on one graph is a version of the knowledge-graph proposition, that heterogeneous facts are more useful when integrated under a common schema than when siloed^[9].

The single graph is what closes the loop between tiers. A Tier-3 sparse solve writes its field back as graph attributes, where a later lookup finds it. A digital-twin update, a measured deflection that differs from nominal, invalidates the stored results that depended on the old geometry and marks them for recomputation, so the retrieved physics tracks the real part rather than the drawing. And because tolerances live beside the fields, the cascade can stop as soon as the decision is settled with margin, which is what lets a coarse lookup or formula answer a question that would otherwise seem to demand a solve. Reduced-order and surrogate models are natural residents of this graph: a reduced basis or an operator network is stored once and evaluated at Tier 4 across the whole parametric family it was fitted for^[5,6,7].

6. Same engineering truth from a neuromorphic chip to a workstation

The final claim is about invariance across hardware. The engineering model, the geometry, the governing relations, the tolerances, is defined once and does not change with the platform. What changes is how far down the cascade a given platform can afford to go, and how it executes each tier. A workstation can run the full high-fidelity solver at Tier 4 whenever it wants. A sub-five-milliwatt neuromorphic controller^[10] may be able to afford only Tiers 1 and 2, a lookup against its local slice of the graph and a closed-form evaluation, and must escalate a Tier-3 or Tier-4 request to a better-resourced peer or defer it. Both are answering questions about the identical engineering model. Execution adapts to the energy available; the engineering truth does not.

What makes this more than a slogan is that the retrievable tiers map cleanly onto low-power substrates. A lookup is an associative recall, which is what neuromorphic and in-memory fabrics do at their energy floor, and a closed-form formula is a short arithmetic expression. The expensive, poorly-mapped operations, meshing, assembly, iterative sparse factorization, sit at Tiers 3 and 4, exactly where the cascade already tries hardest not to go. The surrogate tier gives a controlled way to trade fidelity for energy when a full solve is unaffordable. For a projection-based reduced-order model, the standard result is that the reconstruction error is bounded by the tail of the field's singular spectrum,

$$\|u - \hat{u}_r\| \leq C \varepsilon_r, \quad \varepsilon_r = \left(\sum_{i>r} \sigma_i^2 \right)^{1/2},$$

so a platform can pick the basis size r , and thus the run-time cost, against a known accuracy budget rather than against a guess^[5]. Neural operators make the analogous trade for families where a linear basis decays slowly, learning a resolution-independent map from parameters to solution field^[6,7]. In every case the surrogate is a compression of the same governing physics, evaluated at whatever fidelity the body can pay for, not a different model with a different truth.

7. Position and discussion

CadFuture is best read as an architecture claim rather than a new numerical method. Its parts are individually old and well founded: exact solid modeling since the 1970s^[1], the finite element method since the 1940s^[2], surrogate and reduced-order modeling as mature fields^[4,5], neural operators as an active present frontier^[6,7], and digital twins as an established engineering practice^[8]. The contribution asserted here is their composition: one ordered cascade over one graph, so that the cheapest admissible physics answers each query and every answer makes the next one cheaper, with the same model spanning the full hardware range.

Several risks are worth stating plainly. The concentration assumption of Section 3 may not hold for exploratory or highly novel tasks, in which case retrieval saves little. Tier boundaries require an admissibility test, a reliable way to know that a cheap tier's answer is valid within tolerance before trusting it; getting that test wrong is a correctness failure, not merely an efficiency one, and building it is the hard part of the system. Cache invalidation on twin updates is a well-known source of subtle error. And the cross-hardware claim of Section 6 is, at the time of writing, a design target validated in simulation and prototype rather than a measured result on a deployed milliwatt device. These are the places where the position should be attacked.

8. Conclusion

A world model becomes actionable for a body when it can answer physical questions about the world within the body's power budget. Full CAD and simulation answer those questions but are too heavy to run where the body is. This report argued a research position: for the concentrated, parametric, tolerance-bounded query streams that real embodiment produces, the needed physics is mostly retrievable rather than computable. CadFuture organizes that retrieval as one cascade, lookup then formula then sparse solver then model, over one graph that carries geometry, fields, tolerances, twin state, and interaction, and it defines the engineering model once so the same truth runs from a milliwatt chip to a workstation with only execution adapting. The claim is falsifiable and, in its cross-hardware form, still largely a prototype. It rests entirely on open, long-established prior art, and the work ahead is to measure how far the retrievable fraction of embodied physics actually reaches.

References

1. A. A. G. Requicha. Representations for rigid solids: theory, methods, and systems. *ACM Computing Surveys* 12(4), 1980. doi:10.1145/356827.356833.
2. R. Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of the American Mathematical Society* 49, 1943. doi:10.1090/S0002-9904-1943-07818-4.
3. O. C. Zienkiewicz, R. L. Taylor, D. D. Fox. *The Finite Element Method for Solid and Structural Mechanics*, 7th ed. Butterworth-Heinemann/Elsevier, 2014. doi:10.1016/C2009-0-26332-X.
4. J. Sacks, W. J. Welch, T. J. Mitchell, H. P. Wynn. Design and analysis of computer experiments. *Statistical Science* 4(4), 1989. doi:10.1214/ss/1177012413.
5. P. Benner, S. Gugercin, K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review* 57(4), 2015. doi:10.1137/130932715.
6. Z. Li, N. Kovachki, K. Azizzadenesheli, et al. Fourier neural operator for parametric partial differential equations. arXiv:2010.08895, 2020 (ICLR 2021).

7. L. Lu, P. Jin, G. Pang, Z. Zhang, G. E. Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence* 3, 2021. doi:10.1038/s42256-021-00302-5.
8. M. Grieves, J. Vickers. Digital twin: mitigating unpredictable, undesirable emergent behavior in complex systems. In *Transdisciplinary Perspectives on Complex Systems*, Springer, 2017. doi:10.1007/978-3-319-38756-7_4.
9. A. Hogan, E. Blomqvist, M. Cochez, et al. Knowledge graphs. *ACM Computing Surveys* 54(4), 2021. doi:10.1145/3447772.
10. C. Mead. Neuromorphic electronic systems. *Proceedings of the IEEE* 78(10), 1990. doi:10.1109/5.58356.

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-2026-11 · Preprint v1
© 2026 Institute for Physical AI @ BMI.
Released for open scholarly use. No proprietary or novel experimental results
are reported.