THERMODYNAMIC COMPUTING · TOOLCHAIN · INTERFACES

Technical Report TR-2026-47
Survey / Position · Preprint v1.1
28 August 2026

THERMODYNAMIC COMPUTING · TOOLCHAIN · INTERFACES

What Do You Type?

The software toolchain of thermodynamic computing, surveyed across five language ecosystems. The hardware argument is settled enough to fund. The interface question, what a person or an agent actually writes and what comes back, is not, and it is now the binding constraint.

David Jean Charlot, PhD, Dean of Physical AI · The Charlot Lab

Correspondence: contact@physicalai-bmi.org · physicalai-bmi.org
Instrument: the ferrotherm workbench · Source: ferrotherm

14 toolchains surveyed across 5 language ecosystems

encoding overhead measured, not asserted

4 gaps named in our own stack, and closed in v1.1

3 regions this review did not reach

Abstract. Thermodynamic and Ising-style computers are past the point where the hardware argument decides adoption. What decides it now is the interface: what a user writes, and what the machine writes back. We surveyed the toolchain layer in five language ecosystems, using Japanese, Chinese, English, German and Korean sources, and found four distinct interface designs, each coherent and each solving a different problem: a modelling language over a solver-agnostic intermediate form (Japan), a deep-learning framework plugin (China), a close-to-the-metal sampling library (United States), and a submit-and-wait web portal for exact solvers (Europe). They differ in almost every respect except one. **Every one of them returns a sample, and none of them returns anything that says whether to believe it.** That is a tolerable interface for a researcher with a plot and an intolerable one for an autonomous

agent, which cannot act on a number of unknown quality, a limitation the 2026 agentic-optimisation literature has independently named. We then measure the cost of the encoding layer that all of these share. Using our own compiler, a q-state variable costs exactly q spins under one-hot encoding, and the compact alternatives do not rescue it: domain-wall encoding saves spins per variable and loses more than it saves once a constraint touches them, and binary encoding is *refused outright*, because a binary indicator is already a product and a pair of them exceeds what pairwise hardware can express. The q-fold inflation is therefore not an implementation artefact. It is what a binary substrate charges for a non-binary variable, and 2026 multi-state silicon exists to avoid it. We place our own stack in this landscape honestly, including four algorithms it does not have.

1. What is in scope, and what is not?

This report is about the layer between a person and a thermodynamic computer: the library, the notation, the intermediate form, and the object that comes back. It is not a survey of devices. The device literature is well served: the 2022 *Nature Reviews Physics* survey of Ising machines as hardware solvers remains the reference,[1] and the physics has since produced probabilistic SRAM annealers in 28 nm, memristor crossbars, multi-state MOSFET p-bits and million-cell sampling units.[2,3,4,5]

The reason to look at the interface instead is that the interface is where the concept is foreign. A person trained on sixty years of deterministic instruction streams does not lack the physics. They lack a way to say what they want, and a way to read what came back. That is a software problem, and it is the one this report is about.

It continues the Charlot Lab thermodynamic track: *The Energy-First Turn* (TR-2026-25) holds the substrate case, and the thermo topic holds the running position. The instrument for this report is ferrotherm, the Institute's open-source thermodynamic computing stack, which is also one of the subjects surveyed and is treated no more gently than the others in §6.

2. What does each ecosystem ask you to type?

Four designs, and the differences are not cosmetic. Each answers a different question about who the user is.

Japan: a modelling language over a neutral intermediate form

The most developed toolchain layer in the field is Japanese, and it is the only one that does not ask you to write a matrix. **JijModeling** expresses a problem in mathematical notation (decision variables, constraints, an objective) and compiles it; **OMMX** is the neutral interchange form between modellers and solvers; **OpenJij** and **JijZept** execute it, and **PyQUBO** compiles symbolic constraints to QUBO with automatic penalty handling.[6,7] OpenJij ships simulated annealing and simulated quantum annealing samplers in one interface, and solves higher-order models without first reducing them to quadratic form.[6]

This is the design that treats the user as a modeller rather than a physicist, and it is the most widely adopted. That is the strongest available evidence about where the barrier actually sits.

China: a plugin for the framework people already use

Boson Quantum's **Kaiwu SDK** presents a coherent Ising machine through a standard programming interface, and ships **Kaiwu-PyTorch-Plugin**: a package for training and evaluating energy-based neural networks on the optical machine from PyTorch.[8] This is the only interface we located that treats the thermodynamic device as an accelerator inside an existing deep-learning workflow rather than as a separate optimisation service. Whatever one concludes about the hardware, the interface decision is the most AI-native in the survey.

United States: close to the metal, and a simulator

Extropic open-sourced **THRML**, a library for thermodynamic hypergraphical models, explicitly so that algorithms for its sampling units can be written before the hardware ships, and so that anyone with a GPU can reproduce the results.[9] Normal Computing took the same route from the other direction with **thermox**, a JAX simulator for the stochastic differential equations its continuous-variable stochastic processing unit implements.[10,11]

Both are honest about what they are: a way to write algorithms for a machine most users do not have. Neither is a modelling language, and neither is trying to be.

Europe: the portal, and the automatic formulator

Europe's contribution to this layer is older and differently shaped: **Biq Mac**, **BiqCrunch**, **BiqBin**, **McSparse** and the **Spin Glass Server** are web services where an instance is submitted and a result comes back, several of them backed by semidefinite branch-and-bound that returns a *proved optimum* rather than a sample.[12,13,14] There is no SDK because there is no expectation of a loop: the interaction is one instance at a time.

Separately, the Technical University of Munich's **MQT Quantum Auto Optimizer** attacks the formulation step directly, translating an optimisation problem into a quantum-compliant form automatically and dispatching it to annealers or gate-model solvers.[15]

Korea: hardware ahead of its toolchain

KAIST reported a semiconductor probabilistic bit generating at roughly 128 pJ and 260 ns per bit, and demonstrated a probabilistic computing system solving combinatorial optimisation with it.[16] This review did not locate a public SDK or programming interface accompanying that device. That is a statement about what we found, not about what exists.

3. What do they all return?

A sample. Or a set of samples, or a best-of. In the European portals, a proved optimum, which is more, and is the exception that makes the pattern visible.

What none of them returns, in this review, is an object that says how good the answer is. Not a bound on what a better answer could have been. Not a statement of whether the search terminated because it finished or because it ran out. Not a diagnostic that says *this particular* *run should not be trusted*. The number arrives with no error bar and no provenance, and the user is expected to supply the judgement.

For a researcher that is fine. They have a plot, a baseline, and a feel for the instance. The interface is not the weak link, because a human is standing behind it.

4. What does that cost an agent?

It costs the agent the ability to make a decision. An autonomous system that calls a sampler and receives a number has exactly three things it can do next: accept it, spend more, or reformulate, and no information with which to choose among them.

The 2026 literature has arrived at this from the other side. *Formalize, Don't Optimize* argues that language models asked to *write* combinatorial solvers produce plausible heuristics that are wrong, and that the correct division of labour is for the model to formalise and hand off to a real solver.[17] *ORAgentBench* measures whether agents can carry operations research tasks end to end.[18] Both presuppose the handoff. Neither examines what has to come *back* across it for the agent to proceed.

We think that is the open question, and that it is a notation question rather than a physics one. The vocabulary an agent needs is small and entirely constructible today:

- **A bound.** *Nothing better than this exists*, so the gap between it and the answer in hand is the most a further search could win. Zero means stop.
- **A proof flag that can say no.** *Optimal* and *optimal, or else we gave up* must not be the same field, because the first thing gets read and the second thing gets quoted.
- **A self-distrust signal.** A run that collapsed onto one basin should say so in a field, not in a footnote a human might read.
- **A cost.** Joules, so an agent with a budget can plan rather than guess.
- **Typed refusals.** Not *it did not work*, which an agent cannot route around, but *this graph has a cut vertex, so split it into biconnected components*, which it can.

None of that requires new physics. All of it requires deciding that the return value is part of the interface.

5. What does the encoding cost, and does the radix have to be two?

Every toolchain in §2 shares one hidden step. A real problem has variables that take more than two values (a colour, a slot, a machine, a route), and the hardware has spins that take two. Something must encode the first into the second, and that step is where the size of the problem is decided.

We measured it rather than reasoning about it, using the ferrotherm compiler on graph colouring over a ring, which is the canonical q-state problem. Spin counts are what the compiler actually emitted, end to end, including whatever the constraint cost:

Compiled spins for n-node ring colouring with q colours. Native is one site per variable, which is what multi-state hardware would charge.

q	variables	one-hot	domain-wall	binary	native	cheapest / native
3	16	48	48	refused	16	3.0×
4	16	64	80	refused	16	4.0×
5	16	80	112	refused	16	5.0×
8	16	128	208	refused	16	8.0×
8	64	512	832	refused	64	8.0×

Three things in that table are worth separating.

One-hot costs exactly *q*. A q-state variable becomes q spins, with no constant and no slack. The inflation is the full radix.

The compact encodings do not rescue it. In isolation a domain-wall variable costs $q-1$ spins and a binary one costs $\lceil \log_2 q \rceil$, 3 spins for eight states, against one-hot's 8. But that is the cost of *holding* the variable, not of *using* it. Once a constraint touches two of them, domain-wall overtakes one-hot and keeps going: at $q = 8$ it costs 208 spins where one-hot costs 128.

Binary is refused, and the refusal is the argument. The compiler declines with a reason: a one-hot indicator is linear in spins, so a product of two is quadratic and fits pairwise hardware; a binary indicator is *already* a product, so a product of two exceeds degree two. The information-theoretic minimum $\lceil \log_2 q \rceil$ is not merely expensive on pairwise hardware. For any problem that constrains its variables against each other, it is unreachable.

> So the q-fold inflation is not an artefact of a particular compiler. It is the price a binary substrate charges for a non-binary variable, and it is paid in the currency the hardware is scarce in: sites and couplings. A machine with a million spins running a colouring problem at $q = 8$ is a machine with 125,000 variables.

Which makes the radix a hardware question with a 2026 answer. Multi-state probabilistic bits implementing the Potts model were demonstrated in floating-body MOSFETs this year, CMOS-compatible Potts annealing has been built with single-photon avalanche diodes, and the sparsity problem those machines raise is under active work.[3,19,20] A native q-state site removes the encoding step entirely rather than optimising it.

There is a second alignment here that we flag as a hypothesis rather than a result. The alphabet of a spin-1 site is $\{-1, 0, +1\}$, and that is also the alphabet of the ternary weight quantisations now standard in efficient inference. A fabric whose native variable is ternary would hold such a weight with no encoding at all. Whether that is a real architectural convergence or a coincidence of alphabets is not something this report establishes, and we state it as the open question it is.

6. Where does our own stack actually sit?

Placing ourselves in this landscape is only worth doing if the placement can be unflattering, so the gaps come first.

> **Four algorithms ferrotherm did not have, at v1.0 of this report.** Parallel tempering with **isoenergetic cluster moves** is the physics-inspired baseline the literature measures Ising machines against; we

have parallel tempering and not the cluster moves, and a 2026 entropic-reservoir successor exists.[21,22] **Goemans–Williamson rounding** is the only worst-case approximation guarantee in max-cut; we compute the certified semidefinite bound it is built on and do not round it to a solution. **Simulated quantum annealing** ships as a standard sampler in OpenJij and we do not have it. **Higher-order models** are solved natively by OpenJij; we quadratise, and report the ancilla cost, which is honest but is not the same capability.

Correction, v1.1, 28 August 2026: all four are now implemented. The paragraph above is left as written rather than edited away, because a report that quietly updates its own gap list is a report whose gap list means nothing. What changed, and what each is checked against:

Isoenergetic cluster moves, and parallel tempering built on them. The move flips a connected component of the disagreement between two replicas in both at once, and is accepted unconditionally because the pair's energy is preserved *exactly*. That is an equality, so it is asserted as one on every move. Measured against the identical ladder with the move switched off, the advantage **grows with system size**, which is the literature's claim rather than a claim of general speed: at 8×8 both arms tie on all twenty instances, at 16×16 the cluster arm wins 9 and loses 0, and at 24×24 it wins 19 and loses 0 for a mean energy 8.0 lower.

Goemans–Williamson rounding, verified against optima *proved* by branch-and-bound on 24 instances where the hypothesis holds. The guarantee flag is false on most instances anyone cares about, and says so.

Simulated quantum annealing by Suzuki–Trotter, with the transverse field annealed down but never to zero, where the Trotter coupling diverges, and with one slice treated as the classical control rather than as a degenerate case.

Higher-order models solved without quadratising, verified against exhaustive enumeration over 2^{14}, reporting the ancillas the reduction would have cost. This one is **not yet on the C ABI**: it needs a model-*building* surface rather than a solver call on an existing handle, and halfdoing it is how a capability ends up reachable from one place, which is the failure §3 of this report is about. Named here rather than left invisible.

Shipped in ferrotherm 0.30.0. The measurement in §5, the survey in §2 and the finding in §3 are unchanged.

What the stack does have that this survey did not find elsewhere is the return value. A ferrotherm result carries a certified lower bound on the ground energy, a proof flag that is true only when a search tree was exhausted, a population diagnostic that says when its own free-energy estimate should be distrusted, a joules ledger, and refusals that name which of several distinct conditions was hit. Those exist across six surfaces (Rust, C, Python, Zig, Julia, and an HTTP and MCP server) rather than in one.

Two recent results indicate the bound machinery is not decorative. On planar instances the stack computes an *exact* maximum cut at 10,000 spins, where its own branch-and-bound with a certified semidefinite bound proves 76. On the toroidal G-set instances it produces the upper bound the benchmark has never published, since every G-set figure is a best cut *found*, and on G11 that bound equals the long-standing best-known cut of 564, which closes the bracket and settles the instance. We note in the same breath that the G-set best-known table has moved since the reference heuristic paper: G81 has since been reported at 14060.[23]

7. What would an interface have to return?

We do not think the answer is another SDK. Japan's modelling-language design is better than what most projects would build, OMMX is a reasonable neutral form, and re-implementing either would be a waste. The missing piece is smaller and sits underneath all of them: a *result type*.

Concretely, a thermodynamic result that an agent can act on would carry the answer, a bound, a proof status with an explicit not-proved value, a self-distrust field, a cost in joules, and, on failure, a typed reason rather than a message. None of that is specific to a device or a vendor. All of it could be carried in an existing interchange form.

That is the proposal this report leads to, and it is deliberately modest: the field does not need a new notation for problems. It needs one for answers.

8. What this review did not locate

Stated as findings of this review rather than as claims about the world.

- A public SDK or programming interface accompanying the KAIST semiconductor p-bit.[16]
- An interface, in any of the five ecosystems surveyed, that returns a bound or an optimality gap alongside a sample. The European exact portals return proved optima, which is a different and stronger thing, and they are not SDKs.
- A published head-to-head of the toolchain layer itself (formulation effort, result fidelity, and what a caller can do with what comes back) as opposed to solver benchmarks.
- Russian, French and Indian-language sources. This review covered Japanese, Chinese, Korean, German and English material, and three regions were not reached. The Japanese and Chinese findings were the two that most changed our prior, so we do not assume the unreached regions are empty.

References

1. Mohseni, N., McMahon, P. L., Byrnes, T. Ising machines as hardware solvers of combinatorial optimization problems. *Nature Reviews Physics* 4, 363–379 (2022). `GRADE: PEER-REVIEWED SURVEY`
2. A compact digital compute-in-memory Ising annealer with probabilistic SRAM bit in 28 nm for the travelling salesman problem. *npj Unconventional Computing* (2026). `GRADE: PEER-REVIEWED, SILICON-MEASURED`
3. Cheong, et al. Multi-state probabilistic computing using floating-body MOSFETs based on the Potts model. *Advanced Materials* (2026). `GRADE: PEER-REVIEWED, SILICON-MEASURED`
4. Energy-efficient Ising solver implementations in forming-free memristor crossbar arrays. *ScienceDirect* (2025). `GRADE: PEER-REVIEWED`
5. pc-COP: an efficient and configurable 2048-p-bit fully-connected probabilistic computing accelerator. arXiv:2504.04543. `GRADE: PREPRINT, SILICON`
6. *OpenJij — an open-source project towards a unified annealing platform*. Jij Inc. `GRADE: SHIPPED SOFTWARE, DOCUMENTED`

7. JijModeling and the OMMX adapter. OpenJij Book tutorial 003. `GRADE: SHIPPED SOFTWARE, DOCUMENTED`
8. Kaiwu SDK documentation and Kaiwu-PyTorch-Plugin. Boson Quantum. `GRADE: VENDOR DOCUMENTATION`
9. Extropic. From One to One Billion: Torx, Thermalizers, and Z1; THRML. `GRADE: VENDOR PUBLICATION + OPEN SOURCE`
10. Normal Computing. thermox: the first thermodynamic computing simulator. `GRADE: VENDOR PUBLICATION + OPEN SOURCE`
11. Thermodynamic computing system for AI applications. *Nature Communications* (2025). `GRADE: PEER-REVIEWED`
12. Biq Mac Solver. Rinaldi, Rendl, Wiegele. `GRADE: SHIPPED SERVICE, PEER-REVIEWED METHOD`
13. BiqBin: a parallel branch-and-bound solver for binary quadratic problems with linear constraints. *ACM TOMS* (2022). `GRADE: PEER-REVIEWED`
14. McSparse, Universität Bonn; Spin Glass Server, Universität zu Köln. `GRADE: SHIPPED SERVICE`
15. MQT Quantum Auto Optimizer. TU Munich / Politecnico di Torino. arXiv:2406.12840. `GRADE: PREPRINT + OPEN SOURCE`
16. KAIST semiconductor p-bit and probabilistic computing system. KAIST research news; 물리학과 첨단기술 review. `GRADE: INSTITUTIONAL ANNOUNCEMENT + REVIEW ARTICLE`
17. Formalize, Don't Optimize: the heuristic trap in LLM-generated combinatorial solvers. arXiv:2605.12421. `GRADE: PREPRINT`
18. ORAgentBench: can LLM agents solve challenging operations research tasks end to end? arXiv:2606.19787. `GRADE: PREPRINT, BENCHMARK`
19. CMOS-compatible Ising and Potts annealing using single photon avalanche diodes. arXiv:2211.12607. `GRADE: PREPRINT, SILICON`
20. Restoring sparsity in Potts machines via mean-field constraints. arXiv:2602.04200. `GRADE: PREPRINT`
21. Zhu, Ochoa, Katzgraber. Efficient cluster algorithm for spin glasses in any space dimension (isoenergetic cluster moves). `GRADE: PEER-REVIEWED`
22. Cluster moves with an entropic reservoir accelerate low-temperature simulations of three-dimensional spin glasses. arXiv:2605.25872. `GRADE: PREPRINT`
23. Performance report of a heuristic algorithm that cracked the largest G-set Ising problems (G81 cut = 14060). `GRADE: UNREFEREED REPORT; CITED FOR THE RECORD, NOT RELIED ON`

```
Preprint v1.1, 28 August 2026 (v1.0 the same day; the only
change is the correction block in §6, which records that
the four gaps it named were closed rather than removing
the paragraph that named them). The encoding measurements
in §5 were produced with ferrotherm's compiler and are
reproducible from the published source. The survey in §2
```

reflects sources reachable in August 2026 and is
explicitly incomplete in the three regions named in §8.