NEUTRONGYM: PHYSICS-GRADED NEUTRON INSTRUMENT DESIGN FOR LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Designing a scientific instrument tests whether language-model agents can do physics rather than recall it, provided the grading cannot be argued with. We introduce NeutronGym, to our knowledge the first executable environment for neutron instrument design: agents build instruments through validating tools, McStas ray-traces what they build, and a level-resolved ladder grades syntax, runtime, structure and science with no LLM judge. Procedural families supply unlimited instances of a fixed layout whose design parameters the agent must set, with held-out parameter regimes; a curated slice, McStasBench, adds 16 tasks from published instruments behind memorization probes and a sandbox. Seven models reproduce at most 7 of the 16, none retrieves a reference, and none meets an improvement target. The environment also trains. Reinforcement learning on its reward takes Qwen3-8B from 11% to 77% of held-out instances of a family whose targets come from a hidden design (69% at a second seed), past an untrained Qwen3-32B, and the recipe holds, at one seed each, on three further gated families. The analysis says what that gain is. Without the ladder's partial credit it collapses by 60 points. From reward alone the trained model reaches what a classical optimizer reaches, at the agent's simulation budget, only when handed the closed-form physics (77% against 81%, a gap that does not separate at this size), while frontier models still solve 98–99%. Getting a trustworthy result meant failing four task designs that no-model baselines could solve, and we release the probes that found them.

1 INTRODUCTION

Designing a scientific instrument asks more of an agent than writing code that runs: it must turn a specification into a physical layout, trade competing constraints against each other, and produce a design whose quality is measured by physics rather than by agreement with a reference answer. Neutron scattering instruments are an unusually clean substrate for this. They are designed in simulation long before construction, the community-standard simulator McStas (Lefmann & Nielsen, 1999; Willendrup & Lefmann, 2020) describes an instrument as a text file of components, a Monte Carlo ray trace returns quantitative observables (intensity, divergence, wavelength distribution) in seconds, and passive neutron optics obey a conservation law that gives the deliverable intensity a computable ceiling needing no reference answer.

Turning such a simulator into an evaluation and training substrate raises three problems that are easy to underestimate. *Grading:* any scalar reward on a simulator can be satisfied in unintended ways; we found nine such failures in our own reward and task generator (§5.1). *Contamination:* many published instruments ship as example files with the simulator, so a model may reproduce a design it has read rather than reasoned about; in a pilot audit, 4 of 7 agent episodes retrieved the reference file through ordinary, permitted tools. *Degeneracy:* a generated family can look discriminating while admitting one fixed answer, or one rule that copies numbers from the prompt, that solves most instances, so its pass rates measure lookup rather than design; we learned this twice (§5.2).

We introduce NeutronGym, an executable, physically verifiable environment for LLM agents designing neutron instruments (§3). Agents act only through 22 validating tools exposed over the Model Context Protocol (Anthropic, 2024); the environment simulates what they build and grades it on a level-resolved ladder (L1 syntax, L2 runtime, L3 structure, L4 science) with no LLM judge. Procedural families supply unlimited design instances with calibrated targets and disjoint held-out

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

parameter regimes; its curated slice, McStasBench (§4), adds reproduction and improvement tasks built from published instruments, 16 scored per model and arm, behind memorization probes, perturbed variants and two never-public held-out instruments. Every scored task fixes the instrument's layout: the benchmark asks for a published layout rebuilt from its specification, the families for the numeric parameters of a fixed layout, and no scored tier asks for a topology choice (the open-design tier is unscored).

The paper follows what happened when we used it. On the benchmark, real instruments can be reproduced from their specifications, but not reliably: the best model, Claude Sonnet 5, rebuilt 7 of the 16 tasks in a single completion and 5 through the tool loop, no episode retrieved a reference, and no agent met an improvement target (§4.2); the open-weights Qwen models (Yang et al., 2025) produced no parseable instrument at all without the tools. Training changed that picture. Reinforcement learning on the environment's reward took Qwen3-8B from 11.3% to 76.7% of held-out instances on a family whose targets come from a hidden design, past an untrained 32B that is no stronger than the 8B there, and the same recipe, at a single seed each, did so on three further gated families (§5.2). The analysis then asks what that gain is. Removing the ladder's partial credit costs 60 points; at the agent's simulation budget a classical optimizer matches the trained model only when handed the closed-form physics, which the model recovered from reward alone; and frontier models still solve nearly every instance (§5.3). Getting a trustworthy version of that result was the hard part: supervised fine-tuning on the model's own successes degraded it three times, and four task designs failed the gate because a no-model baseline solved them with a fixed answer, a rule copying limits from the prompt, or a whole class of designs (§5.1).

Contributions.

- **Tasks worth training on.** An admission protocol for procedurally generated scientific tasks: three no-model probes (the best fixed answer, the best rule that reads the prompt, and every instance's solution applied to every other), decided on a one-sided confidence bound, plus a concentration diagnostic for the class-level shortcuts the bound cannot see. Nine documented reward failures motivated it; four of our own designs failed it.
- **NeutronGym and McStasBench.** An executable environment for neutron instrument design, to our knowledge (search through September 2026) the first, with validated construction, compiled execution at about 30 rollouts per second per CPU core, a level-resolved reward ladder with a reference-free phase-space bound on its maximization families, and calibrated procedural families; and its curated slice of published instruments, held behind memorization probes, never-public held-out instruments, perturbed variants and a sandbox.
- **What training on it shows.** RL takes an 8B model past an untrained 32B on four gated families and, from reward alone, to within an insignificant four points of a classical search handed the closed-form physics; the ladder's partial credit is what makes that trainable, and imitation of the model's own successes degrades it.

2 Related Work

Language-model agents that design physical apparatus against a simulator are no longer new, and the closest systems run the loop we run. Kharel et al. (2026) let agents propose, simulate and revise photonic designs against fabrication and physical-consistency gates, and argue the recipe applies wherever a simulator and acceptance criteria exist; GRACE (Hill & Ryoo, 2026) reproduces and improves Geant4 setups under physics-motivated utility functions; OptiAgent (Geng et al., 2026) trains on a gated multi-level reward for optical design whose top level is an LLM judge, where our ladder has none. Detector design has a simulator reward with RL alone (Qasim et al., 2024) and with LLMs proposing designs that RL then refines (Zoccheddu et al., 2026), and benchmarks of agents on scientific simulators and experimental design are numerous (Gandhi et al., 2025; Kumar et al., 2026; Shen et al., 2026; Chi et al., 2026; Liu et al., 2026a; Chen et al., 2025). Nearer our domain, LLM agents already operate beamlines and analyze scattering data (Chen et al., 2026; van der Vleuten et al., 2025; Li et al., 2026a; Xiao et al., 2026; Robledo et al., 2025; Ding & Do, 2026b), but none designs neutron instruments, whose classical optimizers predate LLMs (Bertelsen, 2017). We do not claim the loop. What neutron optics adds is a conservation law that bounds intensity

with no reference (§3), Monte Carlo observables whose noise forces per-observable tolerances, and matching tasks that no open-weights policy solves without measuring (§5.3).

The training method is likewise standard, and we claim no novelty for it. Rejection sampling (Dong et al., 2023; Zelikman et al., 2022) is the baseline that fails here; simulator- and verifier-grounded RL (Shi et al., 2026; Cai et al., 2026; Zeng et al., 2026) is the method that works. RLVP (Cai et al., 2026) is the nearest: it post-trains with the structure our ladder uses, validity gates and then a continuous physics score, for PDE-solver code, and reports the result types we report, one policy over several families, transfer to a held-out family, and a small post-trained model against a prompted frontier model. On that last comparison our result is the opposite, the frontier models staying ahead (§5.3). Calibrating difficulty to the agent follows Zeng et al. (2026) and Guo et al. (2025a).

What this paper adds sits around the training rather than inside it. Agents exploit tool-use rewards and game verifiers (Thaman, 2026; Pan et al., 2024; Helff et al., 2026), verifiers can be fuzzed (Ray, 2026; Roth et al., 2026), agent benchmarks have been broken outright (Wang et al., 2026), and procedural generation is an established contamination defence (Srivastava et al., 2026; Bentham & Srikumar, 2026). We take those warnings as a design requirement rather than a caveat: every family must survive no-model probes before any pass rate is read as capability, and four of our own designs did not. The contrast with environments whose feedback a language model produces (Liu et al., 2026b) sharpens what a physical simulator buys and costs: our reward cannot hallucinate, because it is a ray trace, but we must ship a real simulator, and the failures we engineered against were degenerate tasks, not fabricated observations. Appendix F lists further related work.

3 The NeutronGym Environment

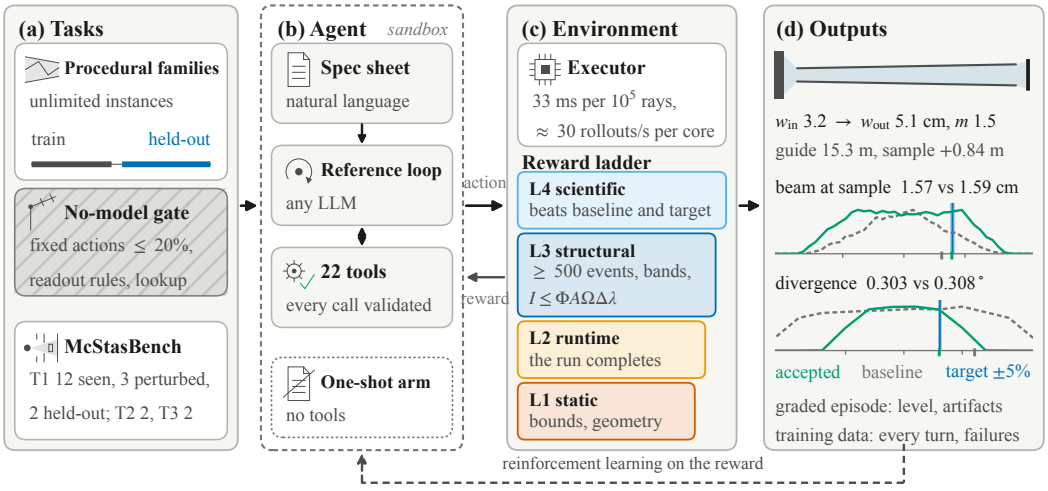

Figure 1: NeutronGym. **(a)** Tasks come from procedural families or from McStasBench, the curated slice; **(b)** the agent works only through validated tools in a sandbox, and **(c)** the environment runs what it builds and grades it on a level-resolved ladder. **(d)** One episode: the instrument the agent arrived at, and the beam profile (horizontal position) and divergence (horizontal angle) McStas returns for it (solid) against the baseline (dashed) and a hidden design's targets, shaded at $\pm 5\%$; m is the supermirror coating index.

3.1 McStas for neutron instrument design

Everything in NeutronGym rests on one fact about the simulator. An instrument in McStas (Willendrup & Lefmann, 2020) is a text file: a beam-ordered list of components (source, optics, sample, monitors), each placed in space and parameterized. McStas compiles the file to C, traces neutron rays through it, and reports what each monitor would see, with Monte Carlo error bars. That report is the reward (Figure 1). No model is ever asked whether a design is good; the environment measures the design the agent actually built, which is why the score cannot be argued with.

The agent never has to write that file and never sees a shell. It works through 22 tools over MCP wrapping McStasScript (Bertelsen, 2026) (Appendix B): it can look up any of the 374 shipped components, place one, run a simulation, and read back per-monitor summary statistics rather than raw arrays. Every mutating call is checked against component metadata as it arrives, so an unknown component or a misspelled parameter fails at the tool call, with the nearest valid alternative suggested, instead of surfacing later as a compiler error. Raw `.instr` import and export remain as an escape hatch.

Trainability comes down to speed. The standard `mcrun` wrapper spends about 2.4 s per run on translation and compilation whatever the ray count, so each family compiles once, keyed on a hash of its comment-stripped source, and every instance is a parameter assignment executed directly on the binary: 33 ms median at 10^5 rays, about 30 rollouts per second per CPU core.

3.2 Grading an action: the reward ladder

An action is graded by how far it gets, not only by whether it works (Figure 1c). Four levels are checked in order, and the first three are engineering. **L1** needs no simulation: the action must assign exactly the family's free parameters, all finite and inside their bounds, and pass the family's geometric specification checks. **L2** asks that a short run at reduced ray count completes. **L3** runs the environment's protocol, with ray count and seed fixed per instance and never chosen by the agent, and requires the graded monitor to hold at least 500 events; on the two maximization families it also requires the observables to stay inside band constraints relative to the baseline and the intensity to respect the phase-space bound below (the matching families carry neither). An instrument that clears L3 exists, runs, and produces usable statistics.

L4 is the physics, and what it asks depends on the family. A *maximization* family wants the figure of merit (FOM) above the instance's calibrated target, $r = \mathrm{FOM}/(\mathrm{baseline} \times \mathrm{target\ fraction}) > 1$ strictly; grading is deterministic at the fixed seed, so resubmitting the baseline scores exactly 1 and is not an improvement. A *matching* family wants every matched observable within its tolerance of the instance's target, with r the tolerance over the worst relative error, so $r > 1$ again means a pass and noisier observables can carry wider tolerances.

The reward follows the ladder: 0.25 per level through L3, then $0.75 + 0.25\min(r, 2)$ once L3 holds, whether or not the bar $r > 1$ is met, capped at 1.25. It rises with depth, then with margin, and that shaping is load-bearing: replacing it with pass/fail costs 60 points (§5.3). Gate-then-score is the structure verifier-grounded post-training uses elsewhere (Cai et al., 2026; Geng et al., 2026).

Phase-space bound. One check needs no reference. Passive neutron optics cannot increase phase-space density (Bertelsen & Lefmann, 2016), so the intensity at a monitor is bounded by $I \leq \Phi A \Omega \Delta \lambda$, source brilliance times the monitor's area, solid-angle and wavelength acceptance, and an action above $1.10\times$ that bound fails L3. It is derived for the two maximization families only (Appendix J); the four matching families, on which every training number is measured, carry none, since their targets are a hidden design's observables and an inflated intensity wins nothing.

3.3 Task families and calibrated targets

A *family* is one instrument layout held fixed while everything numeric varies: some parameters describe the instance the agent is handed, the rest are the design knobs it has to set. An *instance* is a deterministic draw from *(family, split, index)*, and the held-out split draws at least one context parameter from an interval disjoint from training, so a held-out instance is a regime the policy never saw, not a fresh seed. Two of our six families ask the agent to *maximize* a figure of merit. Four ask it to *match* targets: the observables a hidden design produced at the protocol seed, the design itself never shown, with calibration discarding any instance whose targets the baseline already meets. That distinction turns out to decide whether a family can be shortcut without designing anything (§5.1); Appendix I gives each family's parameters, specification, held-out regime and status.

A target means nothing except relative to what is achievable on that instance. We therefore calibrate each with a constraint-filtered search (30 random valid samples, then a local pattern search) that scores every candidate exactly as an agent is scored, at the protocol seed, so target and agent share the same Monte Carlo noise; the target is $0.85\times$ the best figure of merit found. We learned this

the hard way: an earlier version set targets against a deliberately undersized baseline, and untrained models passed 80–93% of the guide family's held-out instances. That was a ceiling, not a task.

4 McStasBench

4.1 Tasks, grading and controls

Where the procedural families are made up, McStasBench asks for real instruments, in three ways. A **T1 (reproduce)** task hands the agent a specification sheet (parameters, component order and types, monitors to report) and asks it to rebuild the instrument; the sheet is generated from a machine-verified reference `.instr`, so it is faithful by construction, and the reference passes its own task at a fresh seed. A **T2 (improve)** task hands over a parameterized baseline and a target 80% of the way to a classical optimum. A **T3 (open design)** task asks for an instrument from scratch and is released unscored. Over 12 published instruments of five classes, 16 tasks are scored per model and arm.

What a pass means deserves stating plainly. Grading runs candidate and reference under one protocol and needs every graded observable (intensity, beam width, spectral centroid and width at the named monitors) inside tolerance, so a T1 pass certifies beam delivery, not class-defining performance such as a diffractometer's resolution. And because the sheet comes from the reference and McStas is deterministic at the graded seed, a faithful rebuild reproduces the reference exactly: 32 of the 39 scored seen-tier passes and all 12 held-out passes match it on every observable exactly. In practice a T1 pass is an exact reproduction, and the tolerance band (Appendix C) rarely decides one.

The harder question is whether a pass is design or recall, and three controls separate the two. Memorization probes record what each model can write from memory before it sees the benchmark; Claude Sonnet 5 reproduces one of the 16 scored references, the others fewer or none, and it fails that task in both arms, so memory accounts for none of its passes. Two held-out instruments have references we authored from published designs and never released. And a perturbed variant of a seen task is kept only on a *pair proof*, in which the perturbed reference passes its own task where the canonical one fails it, so a memorized file fails exactly where a design that follows the sheet passes. Underneath all three sits the sandbox: in benchmark mode the tool server refuses example retrieval and file loads outside the episode, the loop has no shell or file access, and an automated transcript audit (Appendix D) invalidates any episode that touched a reference.

Protocol. All numbers use the NeutronGym reference loop, a model-agnostic chat-completions client over the tool server; a one-shot arm asks for the whole `.instr` file in a single completion, without tools, graded identically. Episodes are pass@1 at temperature 0 with a 50-turn cap, each API model pinned to one serving provider (Appendices C–E).

4.2 A small-scale benchmark test

Table 1: McStasBench results, one episode per task; no per-model difference is significant at this size. "Dominant failure" is where most failed loop episodes stop on the ladder. Maverick's held-out episodes ran on two serving routes (Appendix A) and all four failed.

	Seen (of 16)		Held-out (of 2)		
Model	Loop	1-shot	Loop	1-shot	Dominant failure
Claude Sonnet 5	5	7	2	1	L1 does not compile
Claude Haiku 4.5	4	4	2	0	L4 wrong physics
Gemini 3.6 Flash	4	6	2	1	L0 never finished
Gemini 3.5 Flash-Lite	2	1	1	0	L0 never finished
Llama 4 Maverick	0	2	0	0	L0 never finished
Qwen3-32B	3	0	2	0	L0 never finished
Qwen3-8B	1	0	1	0	L0 never finished

McStasBench is small by construction: 16 scored tasks, one episode per task and arm, seven models (Table 1; Appendix M). That is enough to ask whether a scaffolded model can rebuild a published instrument from a specification and not enough to rank models, since the Wilson 95% interval on 7

of 16 spans 23–67%, and we read it accordingly. Failed episodes are placed on the ladder of §3 by what they produced: **L0** no finished instrument, **L1** does not compile, **L2** dies during the run, **L3** a graded monitor missing or starved, **L4** gradable but outside tolerance.

Three things came out of it. First, the controls held: no episode in any arm retrieved a reference through the tool surface. Second, reproduction is possible but unreliable. The best model passes 7 of 16 tasks one-shot (5 of the 12 instruments) and 5 through the loop (3 instruments), and the Qwen models show what the tools are worth, passing nothing one-shot because they never emit a parseable file. Third, the improvement tier is unsolved: no agent in either arm met the target of the one scored T2 task or of its development twin, each set 80% of the way to a classical optimum that improves the baseline by $2.95\times$ and $1.47\times$, which budget-matched comparisons of LLMs against classical search have found elsewhere (Rodrigues et al., 2026).

The scaffolds fail in different places. Loop and one-shot pass counts do not differ for any model (Fisher exact $p \geq 0.22$), so we make no scaffold claim, but their failures differ in kind: one-shot failures pile up at L1, where the file does not compile, and loop failures at L0, where the agent never finishes an instrument. Over the 14 model–instrument cells of the held-out tier the loop passed 10 against 2 one-shot ($p = 0.006$); those two instruments are also easier than the seen tier, where the loop passed 19 of the seven models' 112 cells ($p < 10^{-4}$), a confound two instruments cannot settle.

5 Training in NeutronGym

Training is where the paper's two halves meet: a task that cannot be shortcut, and a method that learns from its reward. Getting the first took longer. We describe the failures that shaped the tasks, the training that followed, and what bounds the result.

5.1 Making the tasks trainable

We attacked every scored signal before trusting it and found nine ways to score without designing anything (Appendix G). Four changed the environment's design, and two of them set all that follows: a rule that read the answer off the prompt, and a family that a whole class of designs could solve.

The admission gate. The rule we arrived at is a gate. NeutronGym accepts a family only after no-model probes of three kinds: fixed actions, rules that read quantities stated in the prompt (including hand-coded physics inversions), and, for target-matching families, every instance's own solution applied to every other instance. A family fails if the baseline passes anywhere, if a copy-type rule succeeds, or if the best constant's pass share exceeds 20% on the one-sided 95% Clopper–Pearson upper bound (Clopper & Pearson, 1934), not the observed share; the 20% level is a choice, and one admitted family clears it by two points. A physics inversion that passes is reported beside the trained model as a reference arm, since it needs the physics the task tests (Appendix H).

What a fixed-answer gate cannot see: concentration. The gate has a blind spot: it bounds what one *fixed* design can pass, not what a *class* can. Our first bender family passed it at 10.7% while a quarter of its instances fell to any sufficiently transparent design, because their hidden design filtered nothing; that is a class, not a point, so no single candidate ever scored high. We therefore also record the design behind each passing episode and report how concentrated those designs are, a diagnostic read by hand rather than a gate. It exposed the broken family (42 distinct designs in 141 passes, one reused 17 times), while the admitted families read 225 distinct in 230 passes (guide match), 138 in 140 (TOF chopper), 115 in 133 (SANS match) and 63 in 227 (bender, both of whose targets follow from one cutoff wavelength). A class-level criterion is an open problem we name rather than solve.

Rejected designs, and the family we train on. Reinforcement learning on the original guide family took the 8B from 18.0% to 98.7%, above the untrained 32B at 77.3%, and looked like the result we wanted. It was not: the prompt prints the two binding limits, and a no-model rule copying them passes 96.3%, indistinguishable from the trained model (147 and 149 of 150 paired instances, $p = 0.50$). The fixed-action gate could not see it, because the rule reads the prompt. Four designs failed the gate, three discarded and one kept for evaluation only: the undersized-baseline guide

version (§3), this repaired guide family, a scattering family whose readout rule passes 56% (the one kept), and the first bender family. We arrived at *guide match*: its targets are the spot size and divergence of a *hidden* per-instance design, matched within $\pm 5\%$, so nothing in the prompt can be copied and no answer transfers. Its best fixed action passes 4.0% (upper bound 7.7%); a closed-form optics inversion passes 27.7% and is the reference arm.

Training and evaluation protocol. *Imitation* keeps the untrained model's successful episodes and fine-tunes on them with LoRA (Hu et al., 2022), in the style of rejection sampling (Dong et al., 2023). *Reinforcement learning* uses GRPO (Shao et al., 2024) on the environment's reward: the training states are every turn of the untrained model's episodes on 300 training instances, failures included (≈2,810 states); each update samples 8 completions per state, scores them with the ladder, and applies a group-normalized policy gradient with a KL penalty to the frozen base (LoRA rank 32, about 3.5 hours on one A100; Appendix K). Every number below is measured on 300 held-out instances of a family (150 for the out-of-range split) at temperature 0 with a 10-turn budget, paired with the baseline on the same instances by exact McNemar (McNemar, 1947) (Appendix L), on a slice that played no part in choosing tolerances or, except where we say so, checkpoints; tolerances were fixed by measuring the untrained model before training (Appendix I).

5.2 Results

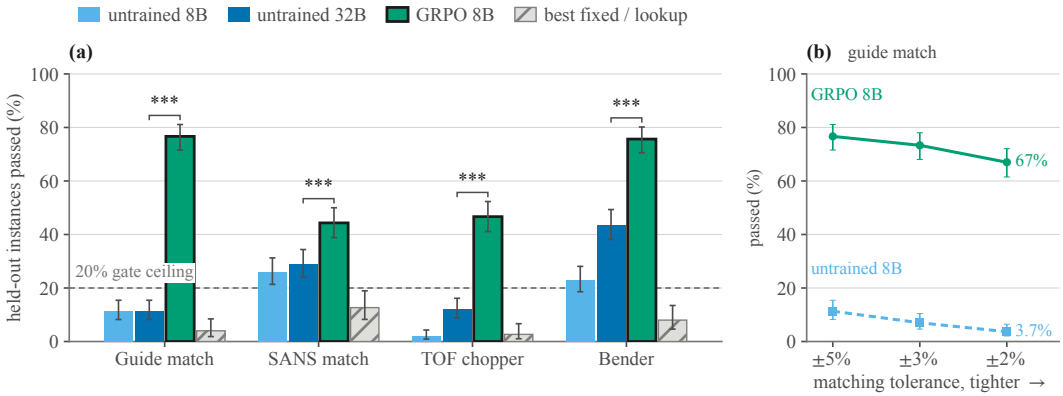

Figure 2: Pass rates on held-out instances ($n = 300$; 150 for the no-model probe) with Wilson 95% intervals, covering instance sampling for a fixed policy, not training-seed variation (8 points) or grading-seed fragility (§5.3). **(a)** The trained 8B passes the untrained 32B on all four gated families and no fixed-answer or lookup baseline comes close; ***: $p < 10^{-3}$, paired exact McNemar against the untrained 32B (discordant 208 vs. 12, 89 vs. 43, 116 vs. 12, 125 vs. 29). Guide match is replicated at a second seed; the other three are single runs. **(b)** Guide match under a tightening tolerance: the trained policy's designs sit inside the tolerance, the untrained one's on its edge.

The headline. On guide match, GRPO takes Qwen3-8B from 11.3% to 76.7% of 300 held-out instances (34 to 230 passes; discordant 213 vs. 17 against the untrained 8B, 208 vs. 12 against the untrained 32B). A second training seed gives 69.0%, and the two seeds differ ($p = 0.043$).

Imitating your own successes does not work. The pre-registered imitation run made the model *worse*: 40.3% to 31.3% on 300 paired held-out instances (discordant 20 vs. 47, $p = 0.0013$), and two further runs on the scattering family regressed as well (Table 9). Cloning your own passing turns teaches a move that was right in the states you already handled and is a trap everywhere else; on-policy RL trains on decisions from every state, failures included.

The recipe holds on three further families, at one seed each. We built three more matching families and gated each before training: *SANS match*, the collimation of a small-angle neutron scattering instrument, matching beam widths at the sample and the beamstop; *TOF chopper*, which leaves geometry altogether, a time-of-flight chopper pair's frequency, phase and opening angle set

to deliver a required mean wavelength and spread; and *bender*, a curved guide that transmits long wavelengths and suppresses short ones, with targets on the transmitted band's centre and width.

Paired on one slice against the untrained 32B, the trained 8B wins every family at $p \leq 8 \times 10^{-5}$ (Figure 2a). We do not average the four: gains against the untrained 8B run from 18 to 65 points, three of the four families rest on a single training run, and SANS match is clean by a slim margin, its best fixed design passing 12.7% (95% bound 18.0%) against the 20% ceiling. On that family, the only one where two checkpoints were scored, the kept checkpoint read 51.0% on the slice used to choose it and 44.3% on a slice chosen for nothing, the number Figure 2 reports.

One policy can cover all four families, but the recipe is unstable. A single LoRA trained from the base model on the pooled decision points of all four families, at one specialist's budget of 220 updates split four ways, reaches 43–64% on every held-out family (Table 11). The per-family pattern should not be read, however. Repeating the three-family configuration with nothing changed but the seed produced **2.3% on TOF chopper, where the untrained rate is 2.0%, against 55.0% for the first run**, and training telemetry alone shows why: about a quarter of sampled groups carried any gradient signal, against roughly half in the run that worked (Appendix N). That is a negative result about multi-task RL, and since its cause lies in the optimization rather than the pooling, it also bounds our confidence in the single-seed specialists.

5.3 Analysis

Table 2: Attempts to explain the result away. On guide match the RL-trained 8B passes 76.7% of 300 held-out instances; each row is a candidate account of that number, read against the reference in its own row (Δ in points). All search rows are coordinate descent, the best of the three optimizers we tried (Nelder–Mead, random search, coordinate descent) at both budgets, and differ only in where the search starts; the seeded search's 4-point lead does not separate at this size (§5.3).

	Family / slice	Pass rate	Read against	Δ
Something simpler did it				
Search from the baseline, agent's budget	guide match	13.3%	76.7% RL	−63
Search from a random start, agent's budget	guide match	14.3%	76.7% RL	−62
Search from the baseline, 3× the budget	guide match	73.7%	76.7% RL	−3
Hand-coded physics rule, no search	guide match	27.7%	76.7% RL	−49
Physics rule, then search, agent's budget	guide match	81.0%	76.7% RL	+4
Frontier models, zero-shot	guide match, first 100	98–99%	76.7% RL	+22
It is an artifact of the evaluation conditions				
Beyond the training parameter ranges	guide match, $n = 150$	45.3%	17.3% untrained	+28
At a $\pm 2\%$ tolerance	guide match	67.0%	3.7% untrained	+63
A family it never trained on	guide (rejected)	34.0%	18.0% untrained	+16
What does account for it				
Reward ladder removed, pass/fail only	guide match	16.7%	76.7% with the ladder	−60

Table 2 collects every attempt we made to explain the result away: something simpler did it, the conditions flatter it, or the ladder did it. This section reads them in turn.

On guide match, reward alone reaches what the physics supplies by hand. Over one held-out slice the arms run from the untrained open-weights models through classical search from the baseline and a hand-coded optics inversion to the trained model and a search seeded by that inversion, with frontier models at 98–99% on the slice's first 100 instances, where the trained model scores 76 (Appendix O). Training recovers about three quarters of the gap to the frontier and does not close it; the frontier models are zero-shot where the trained model saw 300 training instances. Coordinate descent started from the closed-form inversion reaches 81.0% at the agent's budget against the trained model's 76.7%, a gap that does not separate at this size (discordant 59 vs. 46, $p = 0.24$), while from a random valid start it reaches 14.3%, no better than from the baseline ($p = 0.80$): the physics, not the start, carries the information, and the model recovered it from reward alone. The optimizers act on the numeric parameters directly and the seeded one is handed the physics; the agent reads the parametrization from a task sheet, must relate it to the observables itself, and reaches

the simulator only through tool calls. Nor is the gain recall: every untrained episode reaches L3, so the gain is all at L4; on guide match and TOF chopper no 8B or 32B policy, trained or untrained, passes on the first turn (SANS match: 2 of 133; bender: 18 of 227); and almost every pass is distinct (225 of 230). The two seeds' pass sets overlap (agreement 0.68, $p = 0.025$).

Stress tests, and one fragility. Tightening the tolerance from $\pm 5\%$ to $\pm 2\%$ widens the trained-to-untrained ratio from $6.8\times$ to $18\times$, and the ordering survives the out-of-range split of Table 2 ($p = 7 \times 10^{-7}$). One fragility counts against the headline: re-grading all 230 passing designs at three fresh seeds, 80% hold at every seed and 88% of re-runs hold, while 5% fail at all three; the misses cluster at the boundary, tolerance-edge fragility rather than wrong designs.

The reward ladder is load-bearing. Replacing the level-resolved reward with pass/fail only, all else fixed, collapses the result to 16.7% (discordant 196 vs. 16, $p = 1.5 \times 10^{-40}$), and the mechanism is visible during training: with sparse reward a median of 0.33 of the 8 sampled groups per update contained any reward spread, against 5–7 with shaping, so nearly every group contributed no gradient. Dense partial credit is what makes this task reliably trainable at 8B scale in our runs, with three bounds: the ablation is one seed against one; the same gradient-starved signature appeared with the ladder present in the collapsed joint run (§5.2), so the ladder is necessary, not sufficient; and pass/fail removes level and margin credit together, so the ablation cannot apportion the 60 points.

The gain partly transfers, and more training is not better. On the rejected guide family (§5.1), which it never trained on, the guide-match model nearly doubles the untrained rate (discordant 76 vs. 28, $p = 3 \times 10^{-6}$) and its rejections before simulation fall from 73 to 20; since a prompt-copying rule passes 96.3% of that family, this shows the habit RL teaches (propose a valid geometry, then refine), not design skill. Training longer hurts: checkpoints at 120, 220 and 340 updates score 50.0%, 73.0% and 56.7% on another held-out slice as training reward keeps rising.

What was pre-registered, and one measurement defect. Pre-registered: the first supervised run and the matched-compute, frontier and ablation experiments; the rest is exploratory. Before 2026-09-16 the environment reported the per-turn figure of merit to the agent as a fraction of the target but labelled it as relative to the baseline; that defect covers the supervised runs (Table 9) and the guide GRPO run of §5.1, none re-measured, and no matching-family or benchmark number.

6 Limitations and Conclusion

Benchmark scale. The curated slice is small: 16 tasks at one episode each cannot rank models or separate scaffolds; it tests whether reproduction is possible at all. Its two held-out instruments are simpler than the seen median and not facility-validated, so they separate neither novelty from difficulty nor model failure from reference error.

Task coverage. No scored task asks for a topology choice: the benchmark grades beam delivery at the monitors a specification names, and the families set two or three parameters of a fixed layout, where every training number lives. The procedural side is narrow in kind, its four matching families were built after we found the shortcuts, the simulation is idealized (ideal monitors, no background, gravity off), and the probe suite is incomplete by construction: the gate bounds what one fixed answer can pass, not a class, and admitted a quarter-degenerate bender family (§5.1).

Strength of the training evidence. The training results are exploratory: one seed per configuration except two, and a multi-family configuration bimodal across seeds (§5.2). The trained model is an 8B on two- and three-parameter tasks, and §5.3 bounds what it has: below frontier models on the same slice, matched at equal budget by a search handed the physics, weaker outside its training ranges, seed-fragile in a fifth of passes; a second seed per family is the next step.

Conclusion. NeutronGym makes neutron instrument design, at the level of a fixed layout's parameters, executable and gradeable by physics: seven models built instruments in it without retrieving a reference, and RL on its reward lifted an 8B past an untrained 32B on four gated families, three at a single seed. Only no-model baselines separate a task that measures design from one that rewards a shortcut. We release it all.

AI USE STATEMENT

We used an AI coding assistant (Claude, via Claude Code) extensively. It helped implement the tool server, environment, and evaluation harness; run and monitor experiments; analyze results; and draft and revise this manuscript. All code is covered by an automated test suite. Every number in the paper is regenerated by scripts from committed per-episode evidence rather than transcribed by hand, and the authors checked the reported numbers against that evidence. The experimental design, pre-registration decisions and scientific claims are the authors' responsibility. Models from the same provider as the assistant are among those evaluated; all models were evaluated through the same scaffold and grader.

ETHICS STATEMENT

This work involves no human subjects or personal data. Neutron instrument design has no direct dual-use concern beyond that of the published simulator. All models were accessed under their providers' terms.

REPRODUCIBILITY STATEMENT

We release the environment as an installable package with a one-command evaluation of the procedural slice, the benchmark tasks and reference instruments, the pinned run configuration (model identifiers, provider pins, protocol constants), the per-episode evidence for every reported result (grading report, full transcript and the instrument the agent built), the training and evaluation records behind every number in §5.2 and §5.3 with the step-level telemetry that diagnoses the unstable run, and the trained LoRA adapters with checksums. Task families, calibration and the constant-policy gate are deterministic given their seeds; the residual nondeterminism is in serving: temperature-0 decoding under batched inference is not bit-reproducible, and one API model behaved differently on two providers (Appendix A). Three interim claims retracted during the work are listed in the release with the evidence that retracted them: a scaffold-superiority claim that did not survive its statistics, an apparent training gain that a no-model baseline matched, and a task family whose targets a quarter of its instances did not constrain. Anonymized repository: https://anonymous.4open.science/r/NeutronGym-8914.

REFERENCES

R. Aksitov et al. ReST meets ReAct: Self-improvement for multi-step reasoning LLM agent. *arXiv preprint arXiv:2312.10003*, 2023.

Anthropic. Introducing the Model Context Protocol. https://www.anthropic.com/news/model-context-protocol, 2024.

O. Bentham and V. Srikumar. InfiniteScienceGym: An unbounded, procedurally-generated benchmark for scientific analysis. In *Conference on Language Modeling*, 2026.

Mads Bertelsen. The automatic neutron guide optimizer guide_bot. *Nuclear Instruments and Methods in Physics Research A*, 867:195–203, 2017. doi: 10.1016/j.nima.2017.06.012.

Mads Bertelsen. McStasScript: a Python interface to McStas. https://github.com/PaNOSC-ViNYL/McStasScript, 2026. version 0.0.84, accessed 2026-09-24.

Mads Bertelsen and Kim Lefmann. Constraining neutron guide optimizations with phase-space considerations. *Nuclear Instruments and Methods in Physics Research A*, 830:313–324, 2016. doi: 10.1016/j.nima.2016.06.003.

Hassina Bilheux, Ken Herwig, Scott Keener, and Larry Davis. Overview of the conceptual design of the future VENUS neutron imaging beam line at the Spallation Neutron Source. *Physics Procedia*, 69:55–59, 2015. doi: 10.1016/j.phpro.2015.07.007.

P. Cai, U. Utkarsh, A. Edelman, C. V. Rackauckas, and R. Gómez-Bombarelli. Reinforcement learning with verifiable physics: Post-training LLMs with continuous rewards. *arXiv preprint arXiv:2607.10474*, 2026.

Z. Chen et al. ScienceAgentBench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations*, 2025.

Z. Chen et al. An agentic artificially intelligent X-ray scientist. *Nature Machine Intelligence*, 8: 1075, 2026. doi: 10.1038/s42256-026-01261-5.

Y. Chi et al. Frontier-Eng: Benchmarking self-evolving agents on real-world engineering tasks with generative optimization. *arXiv preprint arXiv:2604.12290*, 2026.

W. Chung, Q. Liu, L. Wu, and J. Gonski. Agentic-AI detector co-design and optimization in vertically-integrated differentiable full simulations. *arXiv preprint arXiv:2604.21804*, 2026.

C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

Lijie Ding and Changwoo Do. ColPackAgent: Agent-skill-guided hard-particle Monte Carlo workflows for colloidal packing. *Digital Discovery*, 5:3694–3701, 2026a. doi: 10.1039/D6DD00285D.

Lijie Ding and Changwoo Do. SasAgent: Multi-agent artificial intelligence system for small-angle scattering data analysis. *Journal of Applied Crystallography*, 59(2):578–585, 2026b. doi: 10.1107/S1600576726001299.

Lijie Ding, Jan-Michael Y. Carrillo, and Changwoo Do. ToPolyAgent: AI agents for coarse-grained bead-spring topological polymer simulations. *Digital Discovery*, 5:901–909, 2026. doi: 10.1039/D5DD00471C.

H. Dong, W. Xiong, D. Goyal, Y. Zhang, W. Chow, R. Pan, S. Diao, J. Zhang, K. Shum, and T. Zhang. RAFT: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023.

D. A. Faroughy, S. Palacios Schweitzer, I. Pang, S. Mishra-Sharma, and D. Shih. ColliderBench: Benchmarking AI agents with particle physics analysis reproduction. *arXiv preprint arXiv:2605.13950*, 2026.

K. Gandhi, M. Y. Li, L. Goodyear, A. Bhatia, L. Li, A. Bhaskar, M. Zaman, and N. D. Goodman. BoxingGym: Benchmarking progress in automated experimental design and model discovery. *arXiv preprint arXiv:2501.01540*, 2025.

Y. Geng, L. Sun, Y. Gao, X. Hu, Z. Yi, X. Qian, W. Hu, J. Bai, and K. Wang. OPTI-AGENT: A physics-driven agentic framework for automated optical design. *arXiv preprint arXiv:2602.23761*, 2026.

Jiacheng Guo, Ling Yang, Peter Chen, Qixin Xiao, Yinjie Wang, Xinzhe Juan, Jiahao Qiu, Ke Shen, and Mengdi Wang. GenEnv: Difficulty-aligned co-evolution between LLM agents and environment simulators. *arXiv preprint arXiv:2512.19682*, 2025a.

X. Guo, Y. Li, X. Kong, et al. Toward engineering AGI: Benchmarking the engineering design capabilities of LLMs. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2025b.

L. Helff et al. LLMs gaming verifiers: RLVR can lead to reward hacking. *arXiv preprint arXiv:2604.15149*, 2026.

J. Hill and H. J. Ryoo. GRACE: an agentic AI for particle physics experiment design and simulation. *arXiv preprint arXiv:2602.15039*, 2026.

M. Ho, B. Liu, J. Chen, A. Wang, and L. Qin. Auto-configuring scientific simulators with lightweight coding-agent adapters. *arXiv preprint arXiv:2606.09774*, 2026.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

P. Kharel, A. Khavasi, X. Chen, and T. W. Hughes. Autonomous agentic design for photonics. *arXiv preprint arXiv:2606.00915*, 2026.

V. Kumar, S. Rajput, Mausam, and N. M. A. Krishnan. MDGYM: Benchmarking AI agents on molecular simulations. *arXiv preprint arXiv:2605.08941*, 2026.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, 2023.

K. Lefmann and K. Nielsen. McStas, a general software package for neutron ray-tracing simulations. *Neutron News*, 10(3):20–23, 1999.

Q. Li et al. Rongzai agent: A large language model-based autonomous assistant for Rietveld refinement of neutron diffraction data. *arXiv preprint arXiv:2605.13911*, 2026a.

X. Li, J. Chen, X. Fang, S. Ding, H. Duan, Q. Liu, and K. Chen. OPT-BENCH: Evaluating the iterative self-optimization of LLM agents in large-scale search spaces. *arXiv preprint arXiv:2605.08904*, 2026b.

X. Liu, T. J. Zhang, B. Schölkopf, Z. Jin, and K. Menou. Stargazer: A scalable model-fitting benchmark environment for AI agents under astrophysical constraints. *arXiv preprint arXiv:2604.15664*, 2026a.

Yi Liu, TingFeng Hui, Wei Zhang, Li Sun, Ningxin Su, Jian Wang, and Sen Su. EnvSimBench: A benchmark for evaluating and improving LLM-based environment simulation. *arXiv preprint arXiv:2605.07247*, 2026b.

Z. Liu et al. GEM: A gym for agentic LLMs. *arXiv preprint arXiv:2510.01051*, 2025.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

H. N. Moussa et al. D3-Gym: Constructing real-world verifiable environments for data-driven discovery. *arXiv preprint arXiv:2604.27977*, 2026.

A. Pan, E. Jones, M. Jagadeesan, and J. Steinhardt. Feedback loops with language models drive in-context reward hacking. In *International Conference on Machine Learning*, 2024.

J. Pan, X. Wang, G. Neubig, N. Jaitly, H. Ji, A. Suhr, and Y. Zhang. Training software engineering agents and verifiers with SWE-Gym. In *International Conference on Machine Learning*, 2025.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023.

S. R. Qasim, P. Owen, and N. Serra. Physics instrument design with reinforcement learning. *arXiv preprint arXiv:2412.10237*, 2024.

J. Ray. Before the model learns the bug: Fuzzing RLVR verifiers. *arXiv preprint arXiv:2606.01066*, 2026.

J. I. Robledo, K. Lieutenant, and P. Willendrup. Small angle neutron scattering in McStas: optimization for high throughput virtual experiments. *arXiv preprint arXiv:2501.06054*, 2025.

C. Rodrigues, O. Vas, I. A. DCosta, and N. K. Prabhakaran. When is an LLM worth it for hyperparameter optimization? a budget-matched study on tabular data finds the warm-start is a default configuration, not the model. *arXiv preprint arXiv:2606.21641*, 2026.

A. Roth, A. Samanta, M. Halevy, Y. Levine, and Y. Efroni. Hack-verifiable environments: Towards evaluating reward hacking at scale. *arXiv preprint arXiv:2605.20744*, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Y. Shen et al. SciAgentGym: Benchmarking multi-step scientific tool-use in LLM agents. *arXiv preprint arXiv:2602.12984*, 2026.

Z. Shi et al. MDAgent2: Large language model for code generation and knowledge Q&A in molecular dynamics. *arXiv preprint arXiv:2601.02075*, 2026.

A. Singh, J. D. Co-Reyes, R. Agarwal, et al. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*, 2024.

G. Srivastava, A. Hussain, Z. Bi, S. Roy, P. Pitre, M. Lu, M. Ziyadi, and X. Wang. BeyondBench: Contamination-resistant evaluation of reasoning in language models. In *International Conference on Learning Representations*, 2026.

K. Thaman. Reward hacking benchmark: Measuring exploits in LLM agents with tool use. In *International Conference on Machine Learning*, 2026.

N. van der Vleuten, A. Flores, S. Mathur, M. Rakitin, T. Hopkins, K. G. Yager, and E. H. R. Tsai. EnvTrace: Simulation-based semantic evaluation of LLM code via execution trace alignment—demonstrated at synchrotron beamlines. *arXiv preprint arXiv:2511.09964*, 2025.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

H. Wang, Q. Mang, A. Cheung, K. Sen, and D. Song. How we broke top AI agent benchmarks: And what comes next. Berkeley RDI blog, https://rdi.berkeley.edu/blog/trustworthy-benchmarks-cont/, 2026.

Jinchen Wang, Daye Xu, Juanjuan Liu, Wei Luo, Peng Cheng, Hongxia Zhang, and Wei Bao. Design and construction of the multiplexing cold neutron spectrometer BOYA with double-column Rowland focusing analyzers. *arXiv preprint arXiv:2501.01143*, 2025.

Peter Kjær Willendrup and Kim Lefmann. McStas (i): Introduction, use, and basic principles for ray-tracing simulations. *Journal of Neutron Research*, 22(1):1–16, 2020.

Z. Xiao et al. NeuDiff agent: A governed AI workflow for single-crystal neutron crystallography. *arXiv preprint arXiv:2602.16812*, 2026.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

E. Zelikman, Y. Wu, J. Mu, and N. D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, 2022.

Z. Zeng et al. RLVE: Scaling up reinforcement learning for language models with adaptive verifiable environments. In *International Conference on Machine Learning*, 2026.

T. Zhang, Z. Liu, Y. Xin, and Y. Jiao. MooseAgent: A LLM based multi-agent framework for automating moose simulation. *arXiv preprint arXiv:2504.08621*, 2025.

S. Zoccheddu, S. R. Qasim, P. Owen, and N. Serra. Large language models for physics instrument design. *arXiv preprint arXiv:2601.07580*, 2026.

Y. Zou et al. El agente: An autonomous agent for quantum chemistry. *Matter*, 2025. doi: 10.1016/j.matt.2025.102263.

APPENDIX

The appendix is organised in the order the main text needs it: what the agent can do (A–B), what it is asked to do (C–E), related work in full (F), how we established that the tasks are worth training on and what the phase-space bound is (G–J), how the training and the statistics were run (K–L), and the full result tables (M–O).

A MODELS AND SERVING

Table 3: Models and serving routes. The eighth row ran one seen-tier episode (`HZB_FLEX`, passed) before being cut on cost and is excluded from every reported number.

Model	Access	Serving	Notes
Claude Sonnet 5	OpenRouter	provider-pinned (Google)	
Claude Haiku 4.5	OpenRouter	provider-pinned (Google)	
Gemini 3.6 Flash	OpenRouter	provider-pinned (Google)	
Gemini 3.5 Flash-Lite	OpenRouter	provider-pinned (Google)	
Llama 4 Maverick	OpenRouter	DigitalOcean[‡]	see §4.2
Qwen3-32B	local vLLM	4 GPUs, YaRN $\times$4, 98,304 ctx	thinking off
Qwen3-8B	local vLLM	1 GPU, YaRN $\times$4, 98,304 ctx	thinking off
GPT-5.2 Pro	OpenRouter	OpenAI	one episode (pass), cut on cost

[‡]Google until that route was withdrawn; its held-out loop episodes ran on Google and its held-out one-shot pair on DigitalOcean.

Provider pins are those recorded in the pinned run configuration. The Qwen models were served with thinking disabled in the chat template, the pinned non-thinking baseline configuration; API models were called with no reasoning parameter set, so at provider defaults. Thinking-enabled Qwen arms were not run, which is a limit on the "past an untrained 32B" comparison.

B THE TOOL SURFACE

The 22 tools the agent acts through, wrapping McStasScript (Bertelsen, 2026) over MCP. Every mutating call is validated against component metadata at call time: an unknown component or parameter is refused with the nearest valid alternative rather than surfacing later as a compiler error. In benchmark mode `list_examples`, `get_example` and `load_instr_file` are disabled, which is what closes the retrieval route the pilot audit found.

C TASKS

Table 5 is the full catalogue. Every episode runs at 10^6 rays under the task's fixed seed; "mon/obs" counts the monitor roles compared against the reference and the observables across them. Seen T1 specification sheets are auto-generated from the reference, T2 targets close 80% of the gap from the baseline to the classical optimum, and T3 is released unscored pending an expert rubric. The 16 scored tasks are the seen T1, perturbed and T2 rows less `guide_divergence`, the task the harness was debugged on. Three of the twelve seen references are design studies or idealizations rather than as-built instruments (`ESS_IN5_reprate` is a repetition-rate multiplication study, `HZB_FLEX` models the instrument after its upgrade, `SESANS_Delft` is an idealised instrument). The one scored T2 task is a fixed instance of the SANS collimation family, whose generated instances a readout rule passes 56% of the time (Table 7); no agent passed the fixed instance, so that rule inflates no reported number.

The tolerance rule. Monitors are matched between candidate and reference by role and position rather than by name, and an observable counts as matched within $\max(\text{rtol}\cdot|x_{\text{ref}}|,\ n_\sigma(\sigma_{\text{ref}}+\sigma_{\text{cand}}))$, so a noisy observable is judged against its own error bars rather than a fixed fraction. Five conditions zero an episode's score outright: no instrument produced, a failed translation, a failed compilation

Table 4: The tool surface. Agents never write files or run a shell; these calls are the whole action space.

Tool	Purpose
Introspection: the 374 shipped components	
`list_components`	Browse components by category or search term
`describe_component`	Full parameter table for one component: type, unit, default, required, documentation
`list_examples`	Browse the shipped example instruments
`get_example`	Full source of one shipped example
Construction: validated at call time	
`create_instrument`	Start a new empty instrument
`add_parameter`	Add an instrument-level parameter settable per run
`add_declare`	Declare an instrument-scope C variable
`add_component`	Append a component in beam order, with position and rotation
`set_parameters`	Overwrite parameters on a placed component
`remove_component`	Remove a component; refused if others are positioned relative to it
`get_instrument`	Current state: parameters and components in beam order
`list_instruments`	Instruments in the registry
`validate_instrument`	Static check without a run: required-parameter audit and structural rules
Execution: asynchronous jobs	
`run_simulation`	Compile and run at a chosen ray count and seed
`job_status`	State of a running job, with elapsed time and log tail
`cancel_job`	Kill a running simulation
`get_results`	Summary statistics per monitor (intensity, error, counts, peak, centre of mass, FWHM), never raw arrays
`get_monitor_data`	One monitor in detail, as a rendered plot or a data slice
Search	
`scan_parameter`	Linear scan of one parameter
`optimize`	Classical optimization (scipy, Virtanen et al., 2020) over declared free parameters
Escape hatch	
`load_instr_file`	Import an existing `.instr` file into the registry
`export_instr_file`	Write the generated `.instr` to disk

or run, a graded monitor role absent, or fewer than 1000 events in a graded monitor. Otherwise the score is the fraction of observables inside tolerance, and a pass requires every one of them.

The grading contract. Every T1 task runs at 10^6 rays under seed 1234 and requires at least 1000 events in each graded monitor. The tolerances are fixed per monitor role rather than per task, so no instrument is graded on a bar tuned to it:

Role	Intensity	Shape	Position
`2d` (position-sensitive)	rtol 0.20, 3σ	beam width x, y: rtol 0.30	—
`wavelength`	rtol 0.25, 3σ	FWHM: rtol 0.50	centre of mass: rtol 0.05
`energy`	rtol 0.25, 3σ	FWHM: rtol 0.50	centre of mass: rtol 0.05
`tof`	rtol 0.25, 3σ	—	centre of mass: rtol 0.05

Intensity is the loosest in relative terms and the only observable with a statistical term, because it varies most between correct implementations of the same specification; centre of mass, which applies to the spectral roles, is the tightest at 5%, because an instrument that delivers the wrong mean wavelength, energy or arrival time is wrong whatever its throughput. The position-sensitive role is graded on intensity and beam widths only, and no class-defining performance figure (a diffractometer's resolution, a SANS instrument's Q range) is graded: a pass certifies beam delivery. In practice the band rarely decides one. Sheets are generated from the reference and McStas is deterministic at the graded seed, so a faithful rebuild reproduces the reference's monitors exactly: 32 of the 39 scored seen-tier passes over the seven models and all 12 held-out passes have every graded observable at zero difference from the reference, and the seven passes with any nonzero difference lie on

Table 5: Every McStasBench task. Seen T1 references are McStas shipped examples; held-out references we authored from published designs (Wang et al., 2025; Bilheux et al., 2015) and never released.

Task	Class	Site	Mon/obs	Note
Seen T1: 12 scored				
ESS_IN5_reprate	ToF spectrometer	ESS	3 / 8	
HZB_NEAT	ToF spectrometer	HZB	3 / 8	
ILL_IN6	ToF spectrometer	ILL	3 / 8	
HZB_FLEX	Triple-axis	HZB	2 / 6	
ILL_H10_IN8	Triple-axis	ILL	2 / 6	
ILL_D2B	Powder diffr.	ILL	2 / 6	
PSI_DMC	Powder diffr.	PSI	2 / 6	
RTP_DIF	Powder diffr.	RTP	2 / 6	Reactor TRIGA PUSPATI
ILL_H15_D11	SANS	ILL	1 / 3	
ILL_H512_D22	SANS	ILL	1 / 3	
ISIS_SANS2d	SANS	ISIS	2 / 6	
SESANS_Delft	Spin-echo SANS	Delft	1 / 3	
Perturbed: 3 scored, each kept only on a pair proof				
PSI_DMC_pert	Powder diffr.	PSI	2 / 6	source distance $\times 1.15$; canonical fails on 2d and wavelength intensity
ISIS_SANS2d_pert	SANS	ISIS	2 / 6	source distance $\times 1.15$; same two observables
HZB_FLEX_pert	Triple-axis	HZB	2 / 6	guide curvature $\times 1.4$; canonical fails on energy intensity
Held-out: 2, shown to each model once				
BOYA_CARR	Multiplexing spec.	CARR	2 / 6	single-channel simplification; not facility-validated
VENUS_SNS	ToF imaging	SNS BL-10	2 / 6	not facility-validated
T2 improve: 1 scored				
sans_collimation	SANS optics	—	FOM	two pinhole radii; detector intensity under a beam-width constraint
guide_divergence	Guide optics	—	FOM	three guide parameters; divergence-monitor intensity. Development task
T3 open design: 2, unscored				
compact_sans	SANS	—	floors	$\leq$ 12 m, wavelength CoM 4.5–5.5 Å, brilliance transfer ≤ 1
ballistic_guide	Guide design	—	floors	coating budget

two instruments (HZB_FLEX and its perturbed twin, ESS_IN5_reprate). A task's *mon/obs* count in Table 5 follows from which roles its specification sheet asks for.

Five further tasks are never scored: templateSANS and templateTOF, used for harness bring-up; two SANS pilots (P1, and P3 with a deliberately underspecified sheet); and P2, the memorization control that asks for PSI_DMC from memory without tools.

D Contamination Controls and Protocol

Memorization probes. Each model is asked, without tools and at temperature 0, to write each reference from memory; answers are scored by normalized text similarity and by recall of the component sequence. Claude Sonnet 5 reproduced one of the 16 scored references (ILL_H10_IN8) and one development-split task, Gemini 3.6 Flash one, Haiku 4.5 and Flash-Lite none. Sonnet 5 fails ILL_H10_IN8 in both arms, so its memorized reference is not among its passes. One outcome

changed between runs despite temperature 0, so probes are recorded per model and per date rather than treated as a fixed property of a model.

Perturbed variants. The three kept perturbations are two source distances scaled by $1.15\times$ and one guide curvature by $1.4\times$ (Table 5 names the observables on which the canonical reference then fails). The other nine seen tasks admitted no perturbation that was both detectable (the canonical reference must fail the perturbed task) and valid, the perturbed reference passing its own.

Held-out instruments. BOYA is a single-channel simplification of the multiplexing cold-neutron spectrometer at CARR (Wang et al., 2025), which removes the multiplexing that distinguishes it; VENUS is the SNS time-of-flight imaging beamline (Bilheux et al., 2015). Both references we authored from the published designs, no public model of either exists, and neither is facility-validated. Their specification sheets are not released.

Leak audit. The transcript audit is automated (`leak_audit.py` in the release), the third layer over the disabled tools and the shell-free loop. It scans every tool call in an episode for the example tools and for any file load whose target lies outside the episode's own tree, records refused attempts (the sandbox held) without invalidating the episode, and treats a flagged call whose result is missing from the transcript as a leak, since the sandbox cannot then be shown to have held. A leaked episode is invalid regardless of score. The audit detects retrieval within the episode; memorization is the probes' job above.

Protocol. Episodes run pass@1 at temperature 0 with a 50-turn cap (the procedural-family protocol of §5.1 uses 10 turns) and environment-controlled seeds. Open-weights models run on vLLM (Kwon et al., 2023) with YaRN scaling (Peng et al., 2023) at a 98,304-token context and thinking disabled; Appendix A gives the per-model serving route. Episodes failing before reaching the model are excluded and re-run, which never re-exposes a held-out task because the model never saw it; no episode in the reported matrix needed this (0 infrastructure exclusions over 225 valid episodes). The API spend was $122.16.

E Task Prompts

A T1 specification sheet states the instruction, the instrument parameters with their defaults, the beam-ordered component list with each component's McStas type, position and parameters, the monitor roles the instrument must contain, and the parameter values to run at. No target value appears: the sheet is a reformatted rendering of the reference, so T1 measures faithful reconstruction through the tool surface, not recovery of a hidden design. Sheets run 42 to 345 lines across the twelve seen instruments (median 120). Below is the shortest of them, `ILL_H10_IN8`, with its parameter list and closing run line elided where marked; the four T2 and T3 prompts follow complete, and then the procedural task sheet for the guide-match instance of Figure 1. All are wrapped to the page width, with em dashes and $\pm$ in ASCII, and are otherwise verbatim.

T1 (reproduce), `ILL_H10_IN8`.

```
Rebuild the following TAS in McStas from its specification, run it, and
    report the integrated intensity (with error and event count) on
    every monitor.

Instrument parameters (defaults in parentheses):
- KF (5)
- KI (0)
- QM (0.5)
- EN (0)
- verbose (1)
- WM (0.233)
[... 22 further parameters ...]

Beamline, in order (positions in meters; AT [x,y,z] RELATIVE to the
    named component; parameters as given):
```

```
- Origin: Progress_bar AT [0, 0, 0] RELATIVE ABSOLUTE ROTATED [0, 0, 0]
-     RELATIVE ABSOLUTE
- Source: Source_gen AT [0, 0, 0] RELATIVE ABSOLUTE ROTATED [0, 0, 0]
-     RELATIVE ABSOLUTE
    parameters: radius=0.10, dist=machine_hkl.l1,
        focus_xw=fabs(WM*sin(machine_real.a1*DEG2RAD)), focus_yh=HM,
        E0=machine_hkl.ei, dE=machine_hkl.ei*0.03, I1=0.5874e+13,
        T1=683.7, T2=257.7, I2=2.5099e+13, T3=16.7, I3=1.0343e+12
- SC1: Collimator_linear AT [0, 0, machine_hkl.l1/4] RELATIVE ABSOLUTE
-     ROTATED [0, 0, 0] RELATIVE ABSOLUTE  (WHEN (ALF1 && BET1))
    parameters: xmin=-WM/2, xmax=WM/2, ymin=-HM/2, ymax=HM/2,
        length=machine_hkl.l1/2, divergence=ALF1, divergenceV=BET1
- Guide_out: Arm AT [0, 0, machine_hkl.l1-0.2] RELATIVE ABSOLUTE ROTATED
-     [0, 0, 0] RELATIVE ABSOLUTE

Your instrument must include at least: a 2D position-sensitive monitor;
    a wavelength monitor.
Run with: KF=5, KI=0, QM=1, EN=0, verbose=1, WM=0.233, HM=0.197, RMH=-1,
    RMV=-1, DM=1.807, NHM=15, NVM=15, WA= [... remaining defaults ...]
```

T2 (improve), `guide_divergence`.

```
Improve this neutron guide design.

Baseline instrument (provided as t2_guide_divergence): a 10 m straight
    supermirror guide (entry = exit = 0.012 m square, m = 1.5) starts
    1.5 m from a 10x10 cm source emitting 5.0 +- 0.5 Angstrom; the
    target is a 2x2 cm sample area accepting +-0.5 degrees, 5 cm behind
    the guide exit (Divergence_monitor 'divmon').

Goal: maximize the neutron flux into that acceptance (integrated
    intensity on 'divmon').
You may change ONLY these instrument parameters, within bounds: w_in in
    [0.01, 0.09] m, w_out in [0.01, 0.09] m, m_coat in [1.0, 3.0].
Target to beat: divmon intensity >= 0.001033 n/s at the evaluation
    protocol (ncount 1e+06, fixed seed -- your result is re-verified
    independently).
Report your chosen parameters and the achieved flux with error and event
    count.
```

T2 (improve), `sans_collimation`.

```
Improve the collimation of this SANS instrument.

Baseline (provided as t2_sans_collimation): a pinhole SANS -- 6 m
    collimation with two circular pinholes (r_pin1 = 0.005 m at 3 m,
    r_pin2 = 0.005 m at 6 m), monodisperse-sphere sample, beamstop, and
    a 60x60 cm 2D detector ('detector') 3 m behind the sample; lambda =
    6.0 +- 0.05 Angstrom.

Goal: maximize the scattered intensity on the detector.
You may change ONLY: r_pin1 in [0.002, 0.02] m, r_pin2 in [0.002, 0.02]
    m.
Hard constraint (pattern-integrity band): the detector pattern RMS width
    dX must stay within [4.539, 5.548] cm (+-10% of the baseline
    pattern). This catches both resolution destruction (over-wide
    pinholes smearing the rings) AND direct-beam leakage around the
    beamstop (which floods the center and shrinks dX) -- the beamstop
    must keep shadowing the direct beam.
Target to beat: detector intensity >= 0.06171 n/s at the evaluation
    protocol (ncount 1e+06, fixed seed).
Report chosen radii and achieved intensity with error, events, and the
    measured dX.
```

T3 (open design), `compact_sans`.

```
Design a compact pinhole SANS instrument for a compact
    accelerator-driven neutron source, from scratch.

Requirements:
- Total instrument length (source to detector) <= 12 m.
- Wavelength 5 Angstrom with a velocity-selector-like band of
-     Delta-lambda/lambda approximately 10% FWHM (model the band at the
-     source).
- Must measure scattering from 100 Angstrom monodisperse spheres (use
-     the standard hard-sphere SANS sample model, 1x1 cm, volume
-     fraction 0.1%, contrast 0.6 fm/A^2) with usable Q coverage
-     reaching Qmin <= 0.006 1/Angstrom.
- Include: source, collimation of your choosing, sample, beamstop, a 2D
-     position-sensitive detector sized to your geometry, and a
-     wavelength monitor.
- Assume a source brightness of 1e8 n/s/cm^2/st/Angstrom (state any
-     further assumptions).

Deliver: the instrument (leave it in the registry), your design
    rationale (collimation choice, distances, detector size vs Q range),
    and the flux on sample and detector with errors and event counts at
    ncount 1e6 with a fixed seed.
```

T3 (open design), `ballistic_guide`.

```
Design the neutron transport from a cold source to a distant sample
    position, from scratch.

Requirements:
- Source: 12x12 cm cold source, lambda = 4-6 Angstrom (model with
-     Source_simple or a Maxwellian source; state your brightness
-     assumption).
- Sample position: 30 m from the source; sample area 3x3 cm; the
-     delivered beam must have divergence within +-0.6 degrees (both
-     axes).
- Coating budget: use m > 3 on at most 10 m of total guide length (m <=
-     3 elsewhere) -- justify where you spend it.
- Topology is YOUR choice: straight guide, tapered, ballistic
-     (expanding-straight-focusing), or elliptic; single guide or
-     segmented.
- Include diagnostic monitors after each stage and a divergence monitor
-     plus PSD at the sample position.

Deliver: the instrument (leave it in the registry), the topology
    rationale, and the figure of merit -- flux into the 3x3 cm, +-0.6 deg
    acceptance at the sample -- with error and event count at ncount 1e6,
    fixed seed. Compare at least two topologies and report both numbers.
```

Procedural family, `guide_match-heldout-000300`. Every family sheet has this shape: the fixed context, the free parameters with their bounds and the baseline point, and the targets, which are filled in at reset from the hidden design. The agent is told the parametrization; what it must supply is the mapping from parameters to observables, by measuring.

```
Instrument design task guide_match-heldout-000300.

Shape the beam delivered by a straight supermirror guide. Choose the
    guide entrance width, exit width and coating so that, at the sample
    position d_sample metres after the guide exit, the beam's horizontal
    spot size and horizontal divergence both match this instance's
    targets. Both are reported as the standard deviation of the
    distribution (spot in cm, divergence in degrees).

Fixed geometry for this instance (instrument parameters you must pass
```

```
    but may NOT change):
  src_wh = 0.114613
  L_in = 1.210129
  L_guide = 15.265908
  wl = 5.37935
  dwl = 0.463611
  d_sample = 0.843826

Design parameters you control (bounds inclusive):
  w_in in [0.01, 0.09]  (baseline 0.05)
  w_out in [0.01, 0.09]  (baseline 0.03)
  m_coat in [1.0, 3.0]  (baseline 2.0)

TARGETS for this instance (the episode passes when EVERY quantity is
    within its stated tolerance):
  spot size (horizontal std) on monitor 'psd': (set at reset) +/-5%
  divergence (horizontal std) on monitor 'divmon': (set at reset) +/-5%

Evaluation runs at a fixed protocol (ncount 100000, env-controlled
    seed); iterate however you like, the graded run is the env's.
```

F Extended Related Work

Beyond the works discussed in §2: detector co-design in differentiable simulation (Chung et al., 2026); simulator configuration agents (Ho et al., 2026; Zhang et al., 2025; Zou et al., 2025; Ding et al., 2026; Ding & Do, 2026a); data-driven discovery environments (Moussa et al., 2026); engineering-design benchmarks (Guo et al., 2025b); iterative self-optimization (Li et al., 2026b); budget-matched comparisons of LLMs against classical search (Rodrigues et al., 2026), whose conclusion our improvement tier reproduces; analysis reproduction (Faroughy et al., 2026); general agentic gyms (Liu et al., 2025); and self-training methods related to our failed supervised runs (Singh et al., 2024; Aksitov et al., 2023; Pan et al., 2025).

G Reward Failures Found During Calibration

All nine failures found in the reward and task generator. The first five were caught during calibration and shakedown; findings 6–9 changed the environment's design and are discussed in §5.1.

H The Probe Suite, Family by Family

Every family faces three no-model probes before its pass rates are read as capability: the best *fixed action* (a grid over the free parameters, the baseline, and every probed instance's own solution applied to all others); the best *rule* that reads quantities stated in the prompt, including hand-coded physics inversions; and the *concentration* of a trained policy's own passing designs, which bounds what a whole class of answers can do rather than a single point. A family is admitted only if the one-sided 95% upper bound on the first stays under 20% and no copy-type rule succeeds; a physics inversion is reported as a reference arm, not a disqualification.

Two details decide whether the gate has any power. *Candidate coverage*: a five-level grid over each free parameter reported the SANS family's best constant at 12%, while the true best, lying between grid points, passed 51%, so candidates now come from the grid, the baseline, every probed instance's own calibrated optimum, and a local refinement around the best of them. *Sample size*: maximizing over hundreds of candidates inflates small samples (the guide's best-constant share at the released bar fell from 20% at $n = 25$ to 13.3% at $n = 150$), and against the 20% ceiling, $n = 25$ certifies only at $\leq 1/25$, $n = 150$ at $\leq 21/150$ and $n = 300$ at $\leq 48/300$. A run observing 5/25, whose 95% interval is 7–41%, had earlier been called a pass. For the maximization families the calibrated optimum exceeds the baseline by a median $14.7\times$ (guide) and $10.1\times$ (SANS), so the bar leaves real headroom.

Table 6: The nine reward failures, where each was found, and its fix.

#	Where	What went wrong	Fix
1	T2 calibration	Unconstrained flux maximization collected 672 n/s of direct beam past the beamstop; an upper limit on beam width cannot catch this, because leakage narrows the pattern.	Two-sided band constraints relative to the baseline pattern.
2	T2 calibration	The simulator's Nelder–Mead optimizer ignored parameter bounds (a width escaped to 2.1 m).	Classical baselines are bounds- and constraint-filtered ensembles.
3	T2 calibration	Winner's curse: the best of 30 noisy evaluations fell below its own target at a fresh seed.	Re-verify any selected optimum.
4	Agent shake-down	Matching monitors by event count graded an agent's diagnostic monitor as its detector; correct instruments from two models failed 3/6.	Match monitors by role and position.
5	Phase-space check	A legitimate action read $1.14\times$ the bound on a 250-event monitor; at $10\times$ the rays it read 0.99.	Evaluate the bound only above the statistics floor.
6	SANS family	Maximal pinholes, with no model, passed 8/25 held-out instances, one at 11,992$\times$ the target: unscattered beam leaked around the beamstop and counted as signal. 22 of 35 cached optima were such leaks.	Geometric direct-beam check at L1.
7	Both families	One fixed action passed about half of all held-out instances (guide 52.3%, SANS 51.0%; $n = 300$).	Per-instance specifications and baselines, calibration under common random numbers, and the gate below.
8	Guide family, repaired	Passed the constant-policy gate at every bar, yet a rule that copies the two limits stated in the prompt passes 96.3% of held-out instances.	Report a no-model readout-rule baseline beside every training result.
9	Bender family, first version	Passed the gate at 10.7%, yet on 24.7% of instances the hidden design filtered nothing, so the targets were the source spectrum's own statistics and any transparent design solved them.	Require hidden designs to alter the observable (cut 25–85% of the incident band); diagnose with design concentration.

I Procedural Family Definitions

Table 8 defines the six families, with the parameters the agent controls, the objective or specification it is graded on, the held-out regime, and each family's status after the gate of §5.1.

J The Phase-Space Bound

The check of §3 is derived for the two maximization families and is inactive elsewhere: the environment's bound table has entries for the guide and SANS collimation families only, and every other family passes the check unconditionally. For the guide family the bound is an étendue bound at the divergence monitor: A is the monitor's window, Ω the solid angle of its $\pm\theta_{\max}$ acceptance on both axes, $\Delta\lambda$ the full source band, and Φ the template's unit brilliance. For the SANS family it is a conservation bound through the pinhole chain: nothing downstream can exceed what the second pinhole admits, so A is that pinhole's area, Ω the solid angle the first pinhole subtends from it, $\Delta\lambda$ the source band and Φ the template's brilliance; the bound moves with the agent's radii. Scattering redistributes into 4π and the detector sees far less than the bound, which is why the check is loose there and the band constraints do the fine-grained work. The margin of 1.10 was calibrated on the guide family at 10^5–10^6 rays: legitimate optima read 0.93–0.99 of the bound, and a stress test of 56 corner, boundary and random-extreme actions produced no false flag with a maximum legitimate utilization of 1.016, the excess over 1 coming from the source component's weighting approximation and protocol-level statistics rather than from the optics. Evaluating the bound only above the statistics floor is what keeps Monte Carlo noise from flagging a correct design (Table 6, finding 5).

Table 7: Probes per family, on 150 held-out instances unless noted. Four designs were discarded: an undersized-baseline guide version that predates the gate (§3, not shown), the guide family below, where a rule copying two limits printed in the prompt passed 96.3% of the 300 instances used for training evaluation (98.0% on the 150-instance probe), the SANS collimation family, kept for evaluation only, and the first bender family, where the gate passed but a quarter of instances had a hidden design that filtered nothing, visible only as concentration.

Family	Objective	Fixed / lookup	Readout rule	Verdict
Guide divergence	maximize	13.3%	**96.3%**	rejected
SANS collimation	maximize	14.0%	**56.0%**	evaluation only
Bender, first version	match	10.7%	—	rejected (concentration)
Guide match	match	4.0%	27.7%†	admitted
SANS match	match	12.7%	3.3%†	admitted
TOF chopper	match	2.7%	0.0%	admitted
Bender	match	8.0%	12.7%†	admitted

†Physics inversion, reported as a reference arm. Concentration of passing designs on admitted families, for the checkpoints Figure 2 reports: 225/230 distinct (guide match), 115/133 (SANS match), 138/140 (TOF chopper), 63/227 (bender, one design reused 18 times). The admitted bender is *more* concentrated than the rejected version (28% distinct against 42/141 = 30%): its two observables both follow from one cutoff wavelength, so a low-dimensional solution set is intrinsic there, and what condemned the rejected version was not concentration alone but the quarter of instances whose hidden design filtered nothing. Concentration is a diagnostic, not a gate: no threshold is applied.

Table 8: Procedural families and their status. Specification checks run at L1 before simulation and band constraints at L3, on the two maximization families only; the matching families carry neither and are graded on hidden-design targets. Tolerances were fixed before training by measuring the untrained 8B (for SANS match the sweep read 45.3% at $\pm 3\%$, 24.7% at $\pm 1.5\%$ and 18.0% at $\pm 1\%$, and $\pm 1.5\%$ was adopted); the sweeps are in the release.

Family	Free parameters	Objective / specification	Held-out regime
Guide (rejected)	entrance and exit width $\in [1, 9]$ cm; $m \in [1, 3]$	maximize flux within the divergence limit $m \cdot 0.099°/\text{Å} \cdot \lambda \leq \theta_{\max}$; exit $\leq$ sample width	guide 12.5–16 m (train 6–12)
SANS (evaluation only)	two pinhole radii $\in [1, 20]$ mm	maximize scattered intensity; beam inside the beamstop, $q_{\min} \leq 1/R_{\text{sphere}}$, beam fits the sample	collimation 4.5–6 m (train 2.5–4)
Guide match (admitted)	as guide, without the divergence-limit check; sample 0.1–1.5 m after the exit	match spot size and divergence within $\pm 5\%$	guide 12.5–16 m (train 6–12)
SANS match (admitted)	two pinhole radii $\in [1, 20]$ mm	match beam width at the sample and at the beamstop within $\pm 1.5\%$	collimation 4.5–6 m (train 2.5–4)
TOF chopper (admitted)	chopper frequency 20–60 Hz, phase, opening angle	match mean wavelength and spread at the sample within $\pm 5\%$	chopper separation 9.5–12 m (train below)
Bender (admitted)	radius of curvature 40–550 m; aperture width 2–8 cm; $m \in [1.5, 4]$	match the transmitted band's mean wavelength within $\pm 0.25\%$ and its spread within $\pm 1\%$; the hidden design must cut 25–85% of the incident band	bender 23–30 m (train 12–22)

Bounding the reward by the phase-space ceiling was proposed for classical guide optimization by Bertelsen & Lefmann (2016).

K Training Details

States. A training set is every turn of the untrained model's episodes over 300 training instances of a family, failures included: about 2,810 states for guide match, 8,592 for the pooled three-family set. A state is the full dialogue up to a decision point; the model is trained on the action it takes there, not on a whole episode.

GRPO. Each update samples 8 states and 8 completions per state at temperature 1.0, capped at 96 new tokens per action with prompts truncated at 6,000 tokens, scores each completion with the reward ladder through a simulation server, and applies a group-normalized policy gradient with a KL penalty to the frozen base: the advantage is $(r - \bar{r})/\sigma$ within a group (groups with $\sigma < 10^{-3}$ contribute nothing), each sampled batch is used for one gradient step on the token log-probabilities of its completions, the KL term is the per-token k_3 estimator against the frozen base, and gradients are norm-clipped at 1.0. Optimizer AdamW; seed 20260916; thinking disabled in the chat template. Training runs in two stages: 120 updates at learning rate 10^{-5} with KL weight 0.02, then 100 more at 3×10^{-5} with KL 0.01, 220 updates in all, about 3.5 hours on one A100. The pass/fail ablation and the joint runs use this same schedule, so they differ from the headline run only in the quantity under test. The 120- and 220-update checkpoints are the stage boundaries; the 340-update point in §5.3 continues the second stage for 120 more updates, and that three-point sweep is scored on a second held-out slice, not the headline slice. GPU time for the whole study was not metered; the per-run figure above is the unit.

LoRA. Rank 32, $\alpha = 64$, applied to the attention projections (`q`, `k`, `v`, `o`) and the MLP projections (`gate`, `up`, `down`) of Qwen3-8B. The merged adapter is served through vLLM for evaluation, and the served model is checked against the training tokenization before any number is read from it.

Imitation. The same LoRA shape, trained on the untrained model's own successful episodes as chat dialogues: learning rate 10^{-4}, 2 epochs, about 32,000 tokens per step, sequences up to 8,192 tokens with the prompt masked out of the loss, seed 20260913. A dialogue whose completion would be truncated is dropped rather than clipped, so the model is never trained on half an answer. Table 9 lists the runs.

Table 9: Supervised imitation runs, each read against its own untrained baseline on the pre-repair task version it ran on. All four predate the 2026-09-16 feedback-label fix (§5.3) and none is comparable with the RL rows of Table 2.

Run	Family	Pass rate	Read against	Δ
Imitation of all turns (pre-registered)	guide, pre-repair	31.3%	40.3% untrained	−9
Imitation of passing turns	guide, pre-repair	52.3%	52.3% by one fixed action	0
Imitation of passing turns	SANS, pre-repair	27.0%	31.0% untrained	−4
Same, harder targets	SANS, pre-repair	1.0%	8.3% untrained	−7

L Statistical Methods

Comparisons on the same instances are paired, and we report the discordant counts rather than only a p-value, so the reader can see how much of a difference rests on how few episodes.

- **Paired exact McNemar** (McNemar, 1947) for two policies on one instance slice: every trained-versus-untrained comparison, the tolerance and turn-budget stress tests, the ablation, the transfer result, and the two training seeds against each other (agreement is the share of instances on which both seeds have the same outcome). The statistic is the discordant pair count, and the exact binomial form is used throughout rather than the χ^2 approximation, which is unreliable when discordant pairs are few. The two guide-match seeds are compared unpaired by Fisher's test because the question there is whether the two rates differ, not which instances.

- **Fisher exact** for unpaired 2×2 comparisons, which is how the loop and one-shot arms are compared on McStasBench: the two arms run the same tasks but are separate episodes, not paired trials.
- **Wilson intervals** for the pass-rate error bars in Figure 2, which stay inside $[0, 1]$ at the small rates the untrained arms produce, where a normal interval does not.
- **One-sided 95% Clopper–Pearson upper bounds** (Clopper & Pearson, 1934) for the constant-policy gate. The gate asks whether a family *could* admit a fixed answer passing more than 20% of instances, so it must be decided on the upper end of the interval, not on the point estimate; Appendix H gives the sample sizes this implies.

We report unadjusted p-values. The paper contains more than a dozen tests, so only comparisons at $p < 10^{-3}$ should be read as confirmatory; the ones near 0.01–0.05 (the held-out scaffold comparison, the seed agreement, the two seeds' rates) are reported as observations. The frontier comparison in Appendix O is unpaired, on the first 100 instances of the slice.

M Seen-Tier Results per Model

Table 10 gives the full per-model breakdown behind Table 1: passes and, for every failed episode, the deepest ladder level it reached.

Table 10: Seen-tier results, 16 scored tasks per cell. Failures are counted by the deepest level reached; L0 means the episode ended without a finished instrument (loop) or without a parseable file (one-shot). Levels are read mechanically from the grading report. Over both arms the 185 seen-tier failures are 89 at L0, 73 at L1, 8 at L2, 8 at L3 and 7 at L4.

	Passes		Loop failures					One-shot failures				
Model	Loop	1-shot	L4	L3	L2	L1	L0	L4	L3	L2	L1	L0
Claude Sonnet 5	5	7	1	0	0	9	1	0	1	2	6	0
Claude Haiku 4.5	4	4	4	2	2	4	0	0	0	0	12	0
Gemini 3.6 Flash	4	6	0	1	0	1	10	1	1	2	6	0
Gemini 3.5 Flash-Lite	2	1	1	0	1	5	7	0	0	0	15	0
Llama 4 Maverick	0	2	0	2	0	1	13	0	0	0	14	0
Qwen3-32B	3	0	0	0	0	0	13	0	0	0	0	16
Qwen3-8B	1	0	0	1	1	0	13	0	0	0	0	16

N Joint Multi-Family Training

Joint training pools the recorded decision points of several families into one training set and trains a single LoRA on the mixture, starting from the same base model the specialists start from; no specialist checkpoint is ever warm-started. The budget is one specialist's, 220 updates, split across the families, so one policy and one training run stand against four policies and four training runs.

The failure is quiet rather than loud: mean reward stayed flat near 0.77–0.83 throughout, and only the held-out evaluation revealed it. What does show it during training is the share of sampled groups carrying any gradient (groups whose completions all score alike contribute no advantage), which held near 4.5 of 8 in `joint3` and fell to 2.1 of 8 over the second stage of `joint3rep`.

O Guide Match: All Policies and Baselines

Table 12 lists every arm run on the guide-match headline slice, from the no-model baselines through the trained policies to the frontier models, on which §5.3 draws.

Table 11: Joint runs, all evaluated on held-out instances 300–599 at 10 turns and temperature 0. `joint3rep` repeats `joint3` exactly (same families, same 8,592 states, same budget and hyperparameters) with only the seed changed, and lands at the untrained rate on TOF chopper. Its stage-A checkpoint was already behind (11.0% against `joint3`'s eventual 55.0%), so the two runs diverged early. Only this three-family configuration was ever run twice: there is no evidence that four-family pooling is the stable one. *The SANS-match specialist's 51.0% is read on the slice its checkpoint was selected on; on a slice chosen for nothing the same checkpoint gives 44.3%, the number Figure 2 reports; the untrained-8B row is on the 300–599 slice, where SANS match reads 22.3%, against 26.0% on the selection-free slice Figure 2 draws, so the 18-point SANS-match gain quoted in §5.2 is the figure's $(44.3 - 26.0)$, not this table's. The other three specialist cells are like-for-like with the joint runs.

Run	Pooled	Guide match	SANS match	TOF chopper	Bender
Specialists, one per family	—	76.7%	51.0%*	46.7%	75.7%
`joint3`	3 families	67.7%	24.0%	55.0%	—
`joint4`	4 families	63.7%	43.3%	56.3%	55.0%
`joint3rep`	3, new seed	34.0%	19.3%	2.3%	—
`joint3rep`, stage A		—	—	11.0%	—
Untrained 8B	—	11.3%	22.3%	2.0%	23.0%

Table 12: Guide match, held-out instances 300–599, a slice used for nothing but this measurement. Agents get 10 turns at temperature 0; classical optimizers get the same simulation budget and act on the parametrization directly, which the agent reads from its task sheet and reaches only through tool calls. Italic rows are no-model baselines. Frontier models were run on the first 100 instances of the slice, where the trained model scores 76 of 100.

Policy	Passed	Note
Untrained Qwen3-8B	34/300 (11.3%)	exhausts the 10-turn cap
Untrained Qwen3-32B	34/300 (11.3%)	scale buys nothing here
Best fixed design or lookup	*(4.0%)*	incl. other instances' solutions
Nelder–Mead, 10 simulations	*8/300 (2.7%)*	40.0% at 30 simulations
Random search, 10 simulations	*22/300 (7.3%)*	20.0% at 30
Coordinate descent, 10 simulations	*40/300 (13.3%)*	73.7% at 30
Coordinate descent, random start, 10 sims	*43/300 (14.3%)*	vs. baseline start 34/31, $p = 0.80$
One-shot physics formula	*83/300 (27.7%)*	optics inversion, best of 36
Formula, then coordinate descent, 10 sims	*243/300 (81.0%)*	59 vs. 46 paired against GRPO, $p = 0.24$
GRPO Qwen3-8B (seed 1)	**230/300 (76.7%)**	213 vs. 17 paired against the 8B
GRPO Qwen3-8B (seed 2)	**207/300 (69.0%)**	independent repeat
GRPO, pass/fail reward only	50/300 (16.7%)	ladder ablation
Claude Sonnet 5	99/100 (99%)	$9.69 for both
Gemini 3.6 Flash	98/100 (98%)	