Scoring each decision, not just the loop: testing agents' scientific judgement in simulation discovery loops

Rohit Pillai
School of Engineering
University of Edinburgh
r.pillai@ed.ac.uk

Andreas Hermann
School of Physics and Astronomy
University of Edinburgh

Marcin Kirsz
School of Physics and Astronomy
University of Edinburgh

Thejas Hulikal Chakrapani
School of Engineering
University of Edinburgh

Fengxiang He
School of Informatics
University of Edinburgh

Abstract

In simulation discovery loops the steps run without human involvement, and the decisions between them, which often need expert judgement, are increasingly being handed to large language model (LLM) agents. Whether the agents make these decisions well is not always measured. Molecular simulation is widely used from drug discovery to batteries and chip manufacturing; machine-learned interatomic potentials (MLIPs) have recently started providing the highly-accurate interatomic forces that underpin it, replacing traditional force fields. Our simulation discovery loop of interest is the one that builds an MLIP, which therefore determines how far novel molecular simulations can be trusted. The MLIP loop has five decisions between its steps. We argue that each decision should be scored on its own, against an answer fixed in advance (the key, or ground truth) and checkable by rerunning the simulation, alongside a (human) domain expert and a fixed if-then rule developed by the expert. As early evidence, we have tested the last decision (fit for use) in a simpler form, on a test bed of completed molecular simulations. Using Claude Opus 5.5, we ran 10 independent trials, each starting with blank memory, to judge whether 11 openly available molecular simulations could be 'trusted'. Against the key it averaged 9.3/11, versus 4/11 both for a rule that trusts every simulation and for one that trusts none. We propose this test for all decisions within simulation discovery loops, to find out when agents can be trusted and when they cannot.

1 The position

In October 2026 developers of widely-used scientific simulation codes described how their code is written [CECAM and NCCR MARVEL, 2026]. The lead developer of a major commercial quantum mechanical package stated that "about 70% of our team now uses multi-agent LLM pools to develop software", which "forces us to rely on AI itself for validation and regression testing". Another said validation now separates good physics from "plausible and wrong physics"; a third, referring to an almost fully agent-generated software package, said "the human work was deciding what 'correct' means". Two of the three developers thus assign the coding tasks to LLMs and leave the *decisions* (or choices between automated steps that often need expert judgement) to people. Every decision left to a person costs expert time, and every decision handed to an agent *that gets it wrong* can carry plausible and wrong physics into all downstream simulations. What is missing from this hybrid approach is a data-driven way to design that split, and a measurement of how effective it is in practice.

2026 ELLIS UnConference, Paris. Session: Next Generation of Discovery Loops.

We study that human/agent split in the loop that builds a machine-learned interatomic potential (MLIP) [Unke et al., 2021]. An MLIP predicts the energy of a set of atoms and, from it, the forces between them. A quantum-mechanical calculation (density functional theory, DFT) [Hohenberg and Kohn, 1964, Kohn and Sham, 1965] can compute energies and forces from first-principles and accurately label hundreds to thousands of configurations (arrangements of atoms); the MLIP is fitted to those labels, checked against labels held back from fitting and against known properties, and used to run simulations. The MLIP thus approximates DFT at far lower cost, over time and length scales out of DFT's reach; foundation models (MLIPs pretrained on many systems) cover solids, liquids, interfaces, chemical reactions and proteins [Batatia et al., 2025], for uses from drug discovery to energy storage and semiconductor manufacturing [Wood et al., 2025], so the decisions made in building and using an MLIP have far-reaching effects, as they determine whether (a wide range of complex) molecular simulations can be trusted. Five workflow decisions arise in MLIP development: (1) which configurations to label next, (2) which architecture and hyperparameters to fit, (3) why a check failed, (4) which fix to try, and (5) whether the final MLIP is fit for use (Figure 1 at bottom of page, details in Section 3). Take decision 3: if a check fails, the developer must decide whether the cause was too little training data, a systematic error in the labels themselves, or a test run too briefly to settle; each problem has a different fix. These decisions thus require *scientific judgement*.

It is important to note that even if the overall MLIP development loop ends, the individual decisions that went into it may not be right: a loop can end correctly after a wrong decision because later labels or a refit made up for it, at the cost of extra (expensive) DFT calculations. We propose that agents in simulation discovery loops therefore should be scored on each decision they make, against an answer fixed in advance and checkable by rerunning the simulation, and alongside a human domain expert and a fixed if-then rule written by the expert in advance, each given the same information. Each test item (a *case*) is a completed loop step into which a known fault has been planted (controlled fault injection), or the same step checked to be free of that fault (a control); its right answer (the ground truth) is written into a *key* in advance. Agents, the human expert and the rule access the same files, and a plain-code script with no LLM involvement marks each verdict (the answer given for a case) against the key. The scores would help identify which decisions agents can make reliably and which need a person [Rabanser et al., 2026]. Agents could then take over those decisions one at a time, with downstream checks that simulations stay reliable and expert time is saved.

2 Existing approaches: agents do the work, and agents judge it

Apart from an early study of models reviewing papers with inserted errors [Liu and Shah, 2023], we found no work scoring an agent's verdict on a finished simulation against faults planted with the answer fixed in advance, with checked controls, repeated trials (independent passes, each with blank memory) and a human expert. Scoring steps rather than outcomes is well established: decision theory judges a decision by what was known when it was made, and process supervision scores each step of an LLM's reasoning [Lightman et al., 2023]. Agent systems for science are usually judged by their output [Gottweis et al., 2026].

MLIP agent systems already make these decisions: MLIP Detective handles decision 3, escalating "only the most suspicious cases to human experts" [Okuno et al., 2026], and its authors explicitly ask for a comparison "against random, rule-based, and expert-designed search strategies"; Lang2MLIP handles decision 1, compared only with a manual pipeline and a fixed schedule [Li et al., 2026]; and MLIPilot handles decision 5 by pass/fail thresholds fixed in advance, with no human comparison

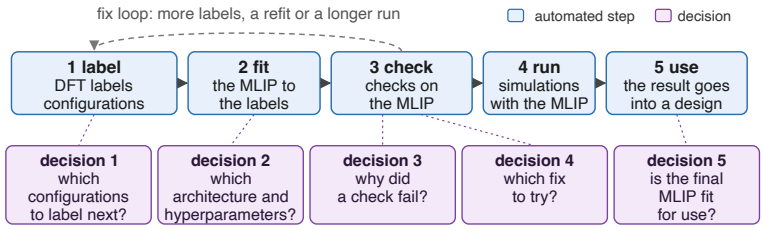

Figure 1: The loop that builds an MLIP, with the five decisions beneath its steps; the steps may be automated, but the decisions require scientific judgement.

[Osaro et al., 2026]. Each system makes its decision, but none of these studies scores that decision against an answer fixed in advance, and no study covers all five decisions.

3 The five decisions in the MLIP development loop

In machine learning terms, an energy-based MLIP is a regression model. Its input is a configuration of N atoms, their element types $Z \in \mathbb{N}^N$ and positions $R \in \mathbb{R}^{N\times 3}$ (and, for a periodic system, the repeating box they fill), often encoded as a graph with the atoms as nodes; its output is the total energy $E \in \mathbb{R}$, with the forces $F = -\partial E/\partial R \in \mathbb{R}^{N\times 3}$ as its negative gradient; the MLIP is fitted to both. Each decision takes what the loop has produced so far (configurations, a search space, a fitted MLIP, check results, a diagnosis) and returns a choice: configurations to label (decision 1), model settings (decision 2), a cause (decision 3), a fix (decision 4) or a verdict on fitness for use (decision 5).

Decision 1, which configurations to label, is the acquisition step of active learning, which picks the training examples expected to improve the model most. It has no key; a fixed acquisition rule (for example, label where the model's uncertainty is largest) is the baseline, and an agent is scored by what it adds over it, such as the same error with fewer DFT labels. Decision 2, the architecture and hyperparameters, can be searched by an algorithm, but in practice experts do this manually [Kirsz et al., 2025]; its key is hard to write, but faults are easy to plant, such as a cutoff radius (the radius defining each local neighbourhood) too small, or a descriptor (the numerical summary of each atom's neighbours) too limited to tell different arrangements apart. Decision 3, why a check failed, is model criticism, and is hard to capture in a fixed rule. For example, a check may fail because an MLIP inherits the known error of the DFT approximation used to label its training data [Lausch et al., 2025, Herrero et al., 2022]; a rule such as "add more data" then fails, as more data labelled the same way cannot correct that misspecification (a systematic error in what the model is fitted to). Decision 4, which fix to try, starts from a diagnosed cause, but one cause can have several fixes: too little data can be met by labelling more configurations near the failure, by a smaller or more constrained (regularised) model, or, if users agree, by narrowing the conditions the potential must cover, at different costs in DFT time. Running the chosen fix tests whether it cures the failure. Decision 5, fit for use, judges downstream utility, whether the MLIP serves its purpose; passing numerical checks alone does not settle that, and in at least one package that judgement is left to the developer [Kirsz et al., 2025].

4 Early evidence from a simplified simulation test bed

We tested decision 5 on 12 finished simulations (cases) from the hands-on course of the NEMD 2026 Summer School [2026], each with its input and output files and a stated purpose, released as a small open test bed, Looks Right [Pillai, 2026]. Each of the course's six measurements was run once with the course's settings and once with one setting changed. Five of the six changes break the simulation for its purpose; the sixth changes the physics instead, and has no fixed answer in the key, which leaves 11 *scored* cases (i.e., with a key fixed in advance). Claude Opus 5.5 was run in ten independent trials and asked to judge whether each case serves its stated purpose. It gave a first verdict and could ask for one rerun with settings changed; a script ran the rerun and marked the agent's final verdict against the key. The test bed contains no MLIP, so only the scoring method carries over to MLIP loops (physics in Appendix A).

Across 11 scored cases, each assessed in ten trials, the agent was wrong on one case in all ten. A setting had been pushed so far that the quantity being measured was no longer defined, as the key stated. In its first verdict, the agent rejected the case as noisy and asked for a rerun. In its final verdict, it trusted the smoother rerun. It named the cause but read it as noise. In another case, the key was wrong. The simulation input inadvertently omitted a term from the formula, leaving the reference value about a tenth too low. In 6 of 10 trials, the agent rejected the case and identified the missing term. Reruns with the term restored confirmed the agent's diagnosis, and the key was corrected. The first case marked a limit of the agent, while in the second the agent corrected its examiners.

Overall, the agent was right on an average of 9.3 of 11 cases per trial, compared with 4 of 11 for a fixed rule that trusts every case as constructed (zero judgement) and 4 of 11 for a rule that trusts none, which never asks for the rerun needed in three cases (Table 1). On nine of the 11 cases the agent gave the same verdict in all ten trials: right on eight, and wrong on one, where a vote across trials [Wang

et al., 2023] would still have been wrong. On two cases, it was right in some trials and wrong in others. One had a wrong answer key. The other was a sound simulation that the agent rejected in 3 of 10 trials. In both, the variation pointed at something worth checking. These preliminary results have limitations: (1) they cover only 11 cases, ten trials, one LLM, one simple system, (2) the planted faults are textbook ones an LLM may recognise from its training data, and (3) no human expert was compared.

5 A proposed test for MLIP development loops

We propose the test in Figure 2 (Appendix B). Each key should be hashed (fingerprinted) and logged before any decider (agent, expert or rule) sees a case. Where a decision has several admissible answers, any of them counts as right. Where a decision has no key, as for decision 1, the fixed baseline takes its place. Faults should be planted in real MLIP builds at saved checkpoints (snapshots of a part-built MLIP), by a person who did not write the fixed rules. Every planted fault should have a control, a checkpoint checked to be free of it, so an agent that rejects everything scores poorly. Some kinds of fault should be kept from the expert who writes the rules. Real mistakes should also be added to the key. Planted faults need not occur at the rates of real ones, so scores on planted and real faults should be reported separately.

Next we propose that three deciders judge the same cases: agents, a domain expert who is not told which cases are faulty, and fixed rules. Each sees the same files for the original case and may ask for one rerun, within the same compute and time limits. The expert writes the fixed rule before any fault is planted. The rule must do more than trust all cases or reject all cases. Each decider names a cause from a short list written with the key, and a script scores both the verdict and the cause, recording when an agent is "right for wrong reasons" [Du et al., 2024] (*shortcut learning*). The script also counts false alarms (rejecting a fault-free control) [Shu et al., 2026] and hand-overs to a person [Wang et al., 2026].

We recommend that each decision be scored over repeated independent trials, and the spread be reported. Each agent and expert verdict includes a probability, allowing us to test calibration (whether stated probabilities match accuracy) [Lin et al., 2022]. We also propose testing whether referring cases on which trials disagree to the expert is worth its cost, against a learned deferral rule (*learning to defer*) [Mozannar and Sontag, 2020]. Every verdict is costed, with penalties for false acceptance, false rejection, reruns, and human minutes set in advance. Per-decision scores may still rank loops wrongly, so the test ends with a downstream check that potentials built by the best-scoring decider reproduce a target property within a specified error using fewer DFT labels and human minutes than by hand.

6 Alternative views and responses

Agents will be capable of everything, so the question is when, not whether. Even so, three limits remain: regulated work requires a named person to sign off some decisions; people must answer for wrong results; and a low-confidence agent must be able to ask for a person. *Planted faults only test the faults their designers thought of, and an LLM may recognise them.* We mitigate this by holding out some kinds of fault, adding real faults and scoring the cause.

7 Future Work

We will run the test of Section 5 on real MLIP builds, varying the agent's harness (the tools, files and reruns available to it), prompt, model (from several developers [Herdağdelen et al., 2026]) and post-training (further training on cases kept separate from the test cases) one factor at a time. Hybrid deciders are a further setting: an LLM paired with Bayesian optimisation and steered by human-written hypotheses has so far been compared by the best result found within a fixed budget [Cissé et al., 2026], and for decisions 1 and 2 it can be scored decision by decision in the same way.

References

Ilyes Batatia, Philipp Benner, Yuan Chiang, et al. A foundation model for atomistic materials chemistry. *The Journal of Chemical Physics*, 163(18):184110, 2025. doi: 10.1063/5.0297006.

CECAM and NCCR MARVEL. Who writes the code? Algorithms, software, creativity, and reliability in the age of AI. Event at EPFL and online, 27 October 2026. Event page with speakers' abstracts, 2026. URL `https://www.cecam.org/lecture-details/who-writes-code-algorithms-software-creativity-and-reliability-age-ai`. Quotations are from the abstracts of G. Kresse, M. Marques and J. Kermode as posted on the event page.

Abdoulatif Cissé, Max E. Cooper, Mengjia Zhu, Xenophon Evangelopoulos, and Andrew I. Cooper. Can we automate scientific reasoning in closed-loop experiments using large language models? *Digital Discovery*, 5(3):1132–1160, 2026. doi: 10.1039/D5DD00520E.

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. Shortcut learning of large language models in natural language understanding. *Communications of the ACM*, 67(1):110–120, 2024. doi: 10.1145/3596490. arXiv:2208.11857.

Juraj Gottweis et al. Accelerating scientific discovery with Co-Scientist. *Nature*, 655(8122):487–496, 2026. doi: 10.1038/s41586-026-10644-y.

Amaç Herdağdelen, Hailey Joren, and Alex Tamkin. Who's aligned to whom? How AI agents shape experts' research judgment calls. Anthropic, 27 October 2026. `https://www.anthropic.com/research/agents-judgment-calls`, 2026.

C. Herrero, M. Pauletti, G. Tocci, M. Iannuzzi, and L. Joly. Connection between water's dynamical and structural properties: insights from ab initio simulations. *Proceedings of the National Academy of Sciences*, 119(21):e2121641119, 2022. doi: 10.1073/pnas.2121641119.

P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Physical Review*, 136:B864–B871, 1964. doi: 10.1103/PhysRev.136.B864.

Marcin Kirsz, Ayobami Daramola, Andreas Hermann, Hongxiang Zong, and Graeme J. Ackland. Tadah! A Swiss Army knife for developing and deployment of machine learning interatomic potentials. *Computer Physics Communications*, 315:109701, 2025. doi: 10.1016/j.cpc.2025.109701. arXiv:2502.02211.

W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Physical Review*, 140:A1133–A1138, 1965. doi: 10.1103/PhysRev.140.A1133.

K. N. Lausch, R. El Haouari, D. Trzewik, and J. Behler. Impact of the damping function in dispersion-corrected density functional theory on the properties of liquid water. *The Journal of Chemical Physics*, 163:034101, 2025. doi: 10.1063/5.0275244.

Wenwen Li, Yuki Orimo, and Nontawat Charoenphakdee. Lang2MLIP: end-to-end language-to-machine learning interatomic potential development with autonomous agentic workflows, 2026. arXiv:2605.14527.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. arXiv:2305.20050.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022. arXiv:2205.14334.

Ryan Liu and Nihar B. Shah. ReviewerGPT? An exploratory study on using large language models for paper reviewing, 2023. arXiv:2306.00622.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *International Conference on Machine Learning (ICML)*, 2020. arXiv:2006.01862.

NEMD 2026 Summer School. Hands-on course materials. Course repository, 2026. URL `https://github.com/Non-Equilibrium-Molecular-Dynamics/NEMD-Summer-School-2026`.

Ryuhei Okuno, Nontawat Charoenphakdee, Kaoru Hisama, and Yuta Tsuboi. MLIP Detective: active failure mode discovery beyond benchmark scores for machine-learning interatomic potentials, 2026. arXiv:2609.08399.

Etinosa Osaro, Santosh Adhikari, Stamatia Zavitsanou, Kelsey Parker, and Dario Rocca. MLIPilot: LLM-driven auto-research for machine-learned interatomic potentials, 2026. arXiv:2605.30889.

Rohit Pillai. Looks Right: an open test of AI agents' scientific judgement in atomistic simulation. Zenodo, version v0.2.2, 2026. URL `https://github.com/RohitWPillai/looks-right`. doi: 10.5281/zenodo.23835985.

Stephan Rabanser, Sayash Kapoor, Peter Kirgis, Kangheng Liu, Saiteja Utpala, and Arvind Narayanan. Towards a science of AI agent reliability. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2602.16666.

Aijie Shu, Bowei Chen, Wenbin Wu, Cathy Yi-Hsuan Chen, and Fengxiang He. DeXposure-Claw: an agentic system for DeFi risk supervision, 2026. arXiv:2606.19501.

Oliver T. Unke, Stefan Chmiela, Huziel E. Sauceda, Michael Gastegger, Igor Poltavsky, Kristof T. Schütt, Alexandre Tkatchenko, and Klaus-Robert Müller. Machine learning force fields. *Chemical Reviews*, 121(16):10142–10186, 2021. doi: 10.1021/acs.chemrev.0c01111.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2203.11171.

Zhiming Wang, Jinwei He, and Feng Lu. Requesting expert reasoning: augmenting LLM agents with learned collaborative intervention, 2026. arXiv:2602.22546.

Brandon M. Wood, Misko Dzamba, Xiang Fu, et al. UMA: a family of universal models for atoms. In *Advances in Neural Information Processing Systems 38*, pages 143528–143564, 2025. doi: 10.52202/085713-4310.

A The test bed in detail

Availability: The test bed, with its key and results, is public [Pillai, 2026]; the course materials are in the summer school's repository [NEMD 2026 Summer School, 2026].

Simulated cases: The cases come from a hands-on course in molecular dynamics on one simple liquid. For each of the course's six measurements there are two simulations, one with the course's settings and one with a single setting changed. Five of the six changes break the simulation for its stated purpose and are scored; the sixth changes the physics and has no fixed answer in the key, so it is left unscored. That gives the 11 scored cases in Table 1.

Answer key: For each scored case the key gives one of three answers: trust; do not trust; or do not trust as is, but trust a named rerun once its output meets a stated criterion. The key was written before the agent saw any case, and one answer was later corrected, as described below. Two of the course's own simulations contain faults: the slip-length simulation (how far the liquid slides along the wall) runs too briefly to settle its answer, and the viscosity simulation leaves out a term, as described below.

Agent's task: In each of ten independent trials the agent (Claude Opus 5.5 in Claude Code, default effort) started with blank memory and was given a case's files (input, output and analysis output), with one sentence saying what the simulation is for. It never saw the key. In its first verdict it answered trust, do not trust, or unresolved, gave its reason in one sentence, and could ask for one rerun. A script ran the rerun, and the agent then gave its final verdict.

Scoring rule: A script with no LLM in it compared each final verdict with the key. An unresolved answer counted as not trusting, and trusting a rerun counted as right only if the rerun's output met the key's criterion.

Case with a wrong key: The quantity measured in the viscosity case is built from two terms, and the course's input left one out. The original key wrongly said trust. In 6 of 10 trials the agent rejected the case and named the missing term. Three new runs with the term restored showed the value was about

a tenth too low, so the key was corrected to do not trust. Under the original key the agent scored 9.1 of 11 per trial; under the corrected key, 9.3.

Case the agent got wrong: In the narrow-pore case one setting was changed so far that the measured quantity is no longer defined, so the key says do not trust. In all ten trials the agent rejected the case as noisy in its first verdict, asked for a rerun, and trusted the rerun in its final verdict. It identified the condition responsible but treated it only as a reason the result would not settle. The course's analysis script prints a warning about that condition, and the agent read it. All ten final verdicts were wrong.

Overall score: Under the corrected key the ten trials give 93 right verdicts out of 110. Of the 17 wrong ones, 14 trusted a case the key says not to trust and 3 rejected one it says to trust. On nine cases the agent was either right in all ten trials or wrong in all ten: right on eight of them, and wrong on the narrow-pore case. On the four cases the key says to trust it was right on 37 of 40 verdicts, and on the other seven, 56 of 70.

Table 1: Right verdicts out of ten trials, per case, against the answer key. "Trust" means the key accepts the case as written for its stated purpose; "do not trust as is" means the key rejects the case as written but allows a named rerun; cases below 10 of 10 are starred; "rerun" marks cases where the agent rejected the original, then trusted the rerun it asked for. In those three cases the key requires the rerun, so rejecting without one counts as wrong; for slip length it accepts either rejecting the case or trusting a longer rerun.

Case	Key	Right of 10	Note
Thermal conductance, as written	trust	10	
Thermal conductance, under-equilibrated	do not trust as is	10	rerun
Density profile, as written	trust	10	
Density profile, coarse bins	do not trust as is	10	rerun
Local viscosity, wide pore	trust	7*	wrongly rejected
Local viscosity, narrow pore	do not trust	0*	agent wrong
Surface tension, as written	trust	10	
Surface tension, overheated	do not trust	10	
Slip length, as written	do not trust as is	10	too short
Viscosity, as written	do not trust	6*	key corrected
Viscosity, fast wall	do not trust as is	10	rerun

B The proposed test as a figure

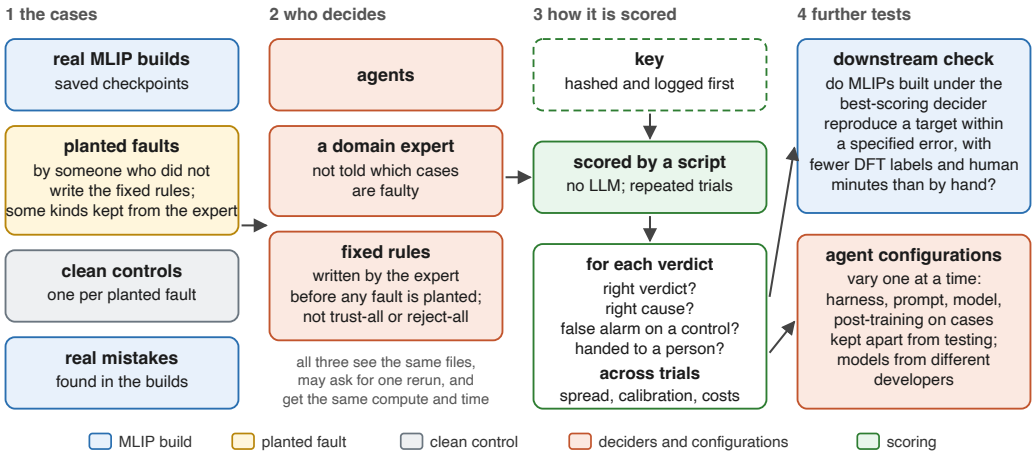

Figure 2: The proposed test of Section 5, adapted to each of the five decisions. The first three columns follow that section; the fourth shows its downstream check and the further testing of Section 7.