Translucent: Language agents for autonomous experimental engineering

Yashas Gunderia
Axcelner
yashas@axcelner.com

Abstract

Models, simulators, automated instruments and agents increasingly support science, yet much work still lies between acquiring experimental evidence and deciding what to test next: interpreting records, establishing what measurements mean, choosing and checking models and defining the experiment. We ask whether a language agent can take responsibility for that preparation and carry an investigation forward through experimental feedback. Translucent runs a general language model, Muse Wonder, that proposes every import, analysis, fit, check, measurement and design request; numerical tools compute, a server enforces approvals and an evidence rule and keeps a persistent record, and a person approves any physical test. In one battery investigation the model turned two unfamiliar cycler exports into a supported fast-charging design after finding that the source cells disagreed and choosing simulated measurements of the target fixture. From raw exports, Translucent recommended in 5/6 runs, all supported and feasible in hidden simulation; the same model in a plain tool loop recommended in 6/6, one unsupported. With eight simulated experiments, Translucent's median charge times were 33.79, 37.17 and 34.13 min versus 42.62, 42.62 and 45.98 min for constrained Bayesian optimization, and every Translucent run beat every run of the same model without experimental feedback, which makes fewer model calls. A prepared numerical procedure matched its model-learning designs with fewer tests. All charging experiments were simulated. In a separate microscopy mode, Muse Wonder directs vision and segmentation models while deterministic code measures and gates findings; we evaluate its checks.

1 Introduction

Between acquiring experimental evidence and deciding what to test next lies preparation that numerical search does not do. Someone reads instrument exports that may describe different specimens or fixtures, decides which columns and segments are comparable, chooses a model and checks that it explains the measurements that matter. Battery fast charging shows the gap. Charging speed is limited by heating and by lithium plating, which depends on the anode potential, so an engineer holding cycler exports from earlier tests must first establish how the fixture at hand heats and cools. Attia et al. accelerated the physical side of this work: early prediction of cycle life and Bayesian selection of experiments shortened the evaluation of charging protocols, using a prepared battery-measurement and prediction pipeline [Attia et al., 2020]. Translucent addresses a different part of the process, preparing and adapting the investigation itself; it does not reproduce Attia et al.'s physical speedup. Later agents generate executable scientific analyses, including Finch's processing of raw cytometry files under expert-specified analysis prompts [Ghareeb et al., 2026], while other systems automate characterization analysis and physical-model construction [Yao et al., 2025, Mandal et al., 2025, Frochte and Schmidt, 2026].

We ask: *within a supplied battery-tool environment, can a general language model turn unfamiliar experimental records into an executable investigation, assembling import mappings, choosing fit and*

held-out observations and revising the investigation from returned evidence, while Translucent enforces target-fixture support and request approvals? Prepared means per-dataset import mappings, segment choices and a prescribed fit-and-check sequence. *Unfamiliar* means received without that preparation, not unseen scientific knowledge. The LG M50 records are public and underlie functions in the simulator's parameter set, one case reformats those same values, and tools and model families are supplied in advance.

The capability under test has five parts. The agent should interpret unfamiliar laboratory records and construct useful analyses; detect when tested fits or cross-fixture predictions fail declared adequacy checks, and revise the investigation using the supplied model families; decide which measurements and experiments are needed next; execute supported computations and revise its investigation from returned evidence; and produce an engineering design only when the evidence rule is met, with the experiment preserved across human interventions. The intelligence comes from the model: the same model in a plain tool loop did much of this work (Section 5.1). Translucent turns that capability into a persistent experimental-engineering system.

Four questions frame the paper. *What could a prepared numerical optimizer not do?* It starts once the problem is prepared; in Section 4, most of the work came before any design search. *What did Muse decide and execute?* It wrote the import mappings and analysis code and chose the data to fit and hold out, the model families, the tests and when to search; numerical routines computed every result. *What did the infrastructure make possible?* Requests ran under approvals bound to their exact content, model revisions persisted with their checks, a test beyond the budget was refused, and the record is checkable offline. *Why might this matter?* In organizations developing physical technologies, preparing each investigation is often the step that waits for a person; a model that does it within a supported environment could work through experiments without that wait. Engineer time was not measured; this is a hypothesis.

Our central contribution is an evaluated end-to-end integration. Starting from raw, unfamiliar records, with no per-dataset mapping or prescribed fit-and-check sequence, a language agent carries one investigation through analysis, model checks, choice of the next test, experimental feedback and a design or a refusal, inside one persistent, enforced experimental record: target-fixture support rules, approvals bound to exact requests, and a campaign history checkable offline. Each ability has antecedents in earlier systems (Section 2); we evaluate their combination in one battery-engineering workflow against the same model without enforcement, prepared numerical procedures and constrained Bayesian optimization. Tools, model families and scientific guidance are supplied, and every charging outcome is simulated. A second, separate capability, multimodal investigation of microscopy video, is evaluated only through its evidence checks (Section 5.4).

2 Related work

Attia et al. [2020] optimize battery cycle life over 224 candidate multistep protocols for a fixed 10 min charge from 0 to 80% state of charge, using early lifetime prediction and Bayesian selection, with physical validation of top protocols on lithium iron phosphate (LFP)/graphite cells; the authors prepared the features, objective and candidate space. Translucent uses similar four-step currents but minimizes simulated charge time under temperature and anode-potential constraints. Lei et al. [2026] use language models to write charging functions or small neural-network protocols evaluated in supplied simulators, reporting gains in a penalized state-of-health metric over Attia's multistep baseline in PyBaMM. LLAMBO inserts language models into Bayesian optimization [Liu et al., 2024]; GOLLuM trains language-model representations through a Gaussian-process objective [Ranković et al., 2026]; PyBOP automates battery fitting [Planden et al., 2025].

Other systems take over parts of the preparation and leave other parts to people. Coscientist

requests measurements and analyzes returned data, with a guiding prompt for spectral interpretation [Boiko et al., 2023]; ChemCrow works under user-instructed permission [Bran et al., 2024]. Robin connects hypotheses to human-executed biological experiments, and Finch's raw-cytometry analyses use expert-specified prompts and protocols prepared by people [Ghareeb et al., 2026]. AILA reads microscope files and writes processing code for user-specified tasks [Mandal et al., 2025]; LLM-RDF fits kinetic models where a request names the fit [Ruan et al., 2024]; SciLink and Modelica-building agents fit physical models to supplied data [Yao et al., 2025, Frochte and Schmidt, 2026]. The Model Discovery Agent expands its hypothesis space after inadequate fits [Murphy, 2026]. Persistent, enforced execution also has antecedents: Lange et al. [2026] gate persisted results with evaluator checks. Autonomous Science Bench evaluates discovery and held-out prediction through a stateful laboratory service with enforced experimental rules and an agent-immutable ledger [Wen and Liu, 2026]. It scores final learning; Translucent instead requires a passing target-fixture check before a recommendation counts as supported.

3 How Translucent turns model choices into an experiment

3.1 Two modes: who decides, who computes, who approves

Translucent has two separate modes, and no evidence passes between them. In *experimental engineering*, the mode of every charging and records-intake study here, Muse Wonder 1.0 [Meta, 2026a] is the only model that plans. An engineer supplies records, specimen identity and an objective; Muse interprets the records, writes import mappings and analysis code, selects fit and held-out data, and requests diagnostics, measurements and designs, each with a stated prediction and reason, revising them from returned results (Figure 1). Translucent's server parses, converts units and explains refusals; deterministic tools fit, simulate and optimize. In *multimodal scientific investigation* of prerecorded microscopy video, Muse Wonder plans the campaign, Muse Spark 1.3 points at objects on gridded frames, the segmentation model SAM 3.1 tracks and outlines the structures at those points, and Muse Spark Open gives an advisory second opinion that never sets a finding's status; deterministic code computes every measurement, and a deterministic gate accepts or refuses each finding (Section 5.4). The rest of this section describes the first mode. A *language agent* here is the model with Translucent's numerical tools, written guidance and loop of requested actions. *Autonomous* means that, after an engineer sets the objective and limits, the agent rather than a person chooses each import, analysis, fit, check and design request that a study offers, under a standing approval for computation; a person must still approve and perform any physical test. *End-to-end* includes that approval and the ingestion of returned measurements.

Written guidance is part of the treatment: recommend only from a numerical search under a model checked on a test it did not fit. Intake instructions add: keep tests out of fitting, and treat source-record fits as evidence rather than proof because the target fixture may differ. Both model arms receive that fixture warning in the objective. The *Unenforced workflow* gives the same model Translucent's parser, numerical tools, simulated lab and planning guidance in a plain tool loop. It states the evidence rule without enforcing it and omits Translucent's import validation, campaign state and intake instructions, so the arms differ in instructions, interfaces and enforcement. *Registered* means criteria fixed in protocol documents before scoring, with amendments disclosed; *frozen* means retained historical runs and outcomes.

3.2 Numerical models, design search and the evidence rule

The battery tools use public LG M50 records [Chen et al., 2020], a Doyle–Fuller–Newman model [Doyle et al., 1993] in PyBaMM [Sulzer et al., 2021], and O'Kane's parameterization with partially reversible

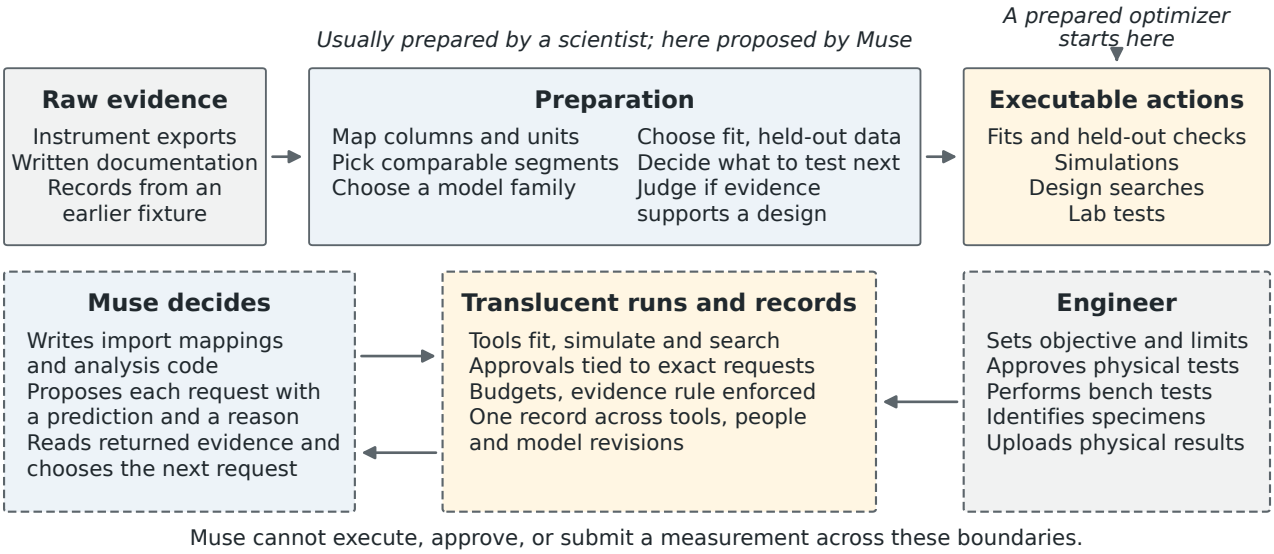


Figure 1: The missing link between raw evidence and executable actions, and who holds which authority. Top: the preparation a prepared optimizer assumes is done. Bottom: Muse proposes; Translucent's tools compute and its server checks approvals, budgets, source bindings and the evidence rule. A person sets limits, alone approves and performs physical tests, and identifies the specimen; checked uploads and computed results return through the server to Muse.

plating [O'Kane et al., 2022]. Four current steps span 10–27.5, 27.5–45, 45–62.5 and 62.5–80% state of charge, followed by constant voltage at 4.2 V if reached. Currents lie on a 0.05C lattice from 0.2C to 1.5C; C-rate is current relative to nominal capacity. Fitting estimates cooling alone or cooling plus a thermal-mass scale, with a nominal 95% region widened for autocorrelated residuals; forecasts evaluate the estimate and up to four extreme points in that region. The design routine uses greedy current reduction and bisection with a full-protocol check. The *Compass-search reference* searches the same lattice under each fixed model, with an unreported evaluation budget; the intake *Hidden-condition reference* applies the design routine with the hidden fixture parameters. Neither establishes global optimality.

Intake support requires an adequate fit and a passing check on target-fixture observations excluded from fitting, with no failed applicable check. Temperature root-mean-square error (RMSE) must be at most $\max(2\sigma, 0.30\ \mathrm{K})$, for declared noise σ. Source-record checks cannot confer support in another fixture. Predicted feasibility additionally requires meeting the objective's constraints. Surface-temperature checks do not validate anode potential or establish physical accuracy.

3.3 Execution and the checkable record

Approval binds to the exact request's content hash, including parameters and prediction. The server enforces budgets, input bindings and software versions. Physical results must match the approved specimen, configuration and units. An insert-only, hash-chained record links observations, models and designs. Offline checking reapplies acceptance rules and recomputes measurements from stored files. This checks internal consistency; physical origin depends on specimen custody. A privileged operator can replace an unanchored history.

4 An investigation from raw records to a design

We follow one scored run in full: Translucent's first repeat of intake case H1 (Section 5.1); it adds no replication. The engineer supplies cycler exports for cells 02 and 03 of the public LG M50 validation set [Chen et al., 2020], their written test procedure and an objective: the fastest 10–80% charge at 25.0°C in the lab's fixture, with peak temperature at most 45.0°C and anode potential at least +25.0 mV. The objective warns that the lab's fixture may differ from the records'. The lab is simulated with hidden fixture parameters; the run may use two lab tests and US$2.50 of model calls. Figure 2 lists every request and its evidence.

Reading the records. Muse inspects both files, runs its own analysis and writes an import mapping. The server refuses four mappings, for schema errors, a time column listed twice and a wrong header row; Muse corrects each. A later attempt to add the second cell under the same dataset name is refused as a conflict, and Muse registers a new dataset holding both cells. Its manifest records what the files cannot establish: no thermocouple calibration, current logged as an unsigned magnitude with direction only in a mode column, a starting state of charge that must be inferred from the documented procedure, and no charge faster than 0.3C or plating measurement. From its segment listing it selects each cell's 1.5C discharge, the largest thermal signal in the records (rises of 27.9 and 25.9 K), fitting cell 02 and holding cell 03 out.

Finding that the records cannot support a design. A cooling-only fit to cell 02 fails the adequacy rule at 1.67 K RMSE with structured residuals; Muse attributes the misfit to the thermal timescale and adds the thermal-mass parameter. Revision 2 fits at 0.19 K. Muse checks it on cell 03, predicting a held-out RMSE of 0.4 K; the check returns 1.82 K. A pooled fit fails at 0.93 K, and fitting cell 03 alone gives 0.15 K. Cooling estimates are 5.224 and 6.161 $\mathrm{W\,m^{-2}\,K^{-1}}$, and mass scales 1.485 and 1.655; their respective nominal 95% regions do not overlap. Muse states that "no pooled record model can support a design" and turns to the fixture where the design would run. The rule requires a passing check on target-fixture tests excluded from the fit; it does not require fitting them. Muse chose both test conditions and, citing the cell disagreement, chose to fit the target test alone instead of checking a record-fitted model.

Measuring the target fixture. Muse requests a uniform 1.5C charge followed by 60 min rest, the largest thermal signal, whose rest constrains the ratio of thermal mass to cooling. It predicts 46.0°C with a stated range [38.0, 54.0]°C; the simulator returns 49.8°C, above the limit. Revision 5 fitted to this test has 0.15 K RMSE. Muse holds out a uniform 1C charge with the same rest, predicting 38.0 [32.0, 44.0]°C; the returned peak is 41.4°C, so the two charges bracket the feasible current, and held-out RMSE is 0.15 K, below 0.30 K. These are recorded predictions, not calibrated probability intervals.

Design and refusal. Only now does Muse request a design search. The numerical search returns 1.5/1.5/1.2/0.8C and 35.88 min, forecasting 44.7°C and +27.5 mV. A request to test that exact design is refused; both lab tests are spent. Muse's conclusion records that the design's 0.255°C temperature margin is small beside the 0.37°C forecast error on the held-out test, so a lab check would be the next experiment, and that anode potential remains a forecast. Post-run hidden assessment finds 35.88 min, 44.7°C and +27.6 mV, matching the Hidden-condition reference's charge time. Revision 5 estimates cooling 5.744 [5.723, 5.758] $\mathrm{W\,m^{-2}\,K^{-1}}$ and mass scale 1.245 [1.240, 1.250], containing the hidden values 5.739 and 1.244. This close recovery partly reflects construction inside the fitted family. All 12 accepted requests use standing computational approval; model spend is US$1.99 of US$2.50.

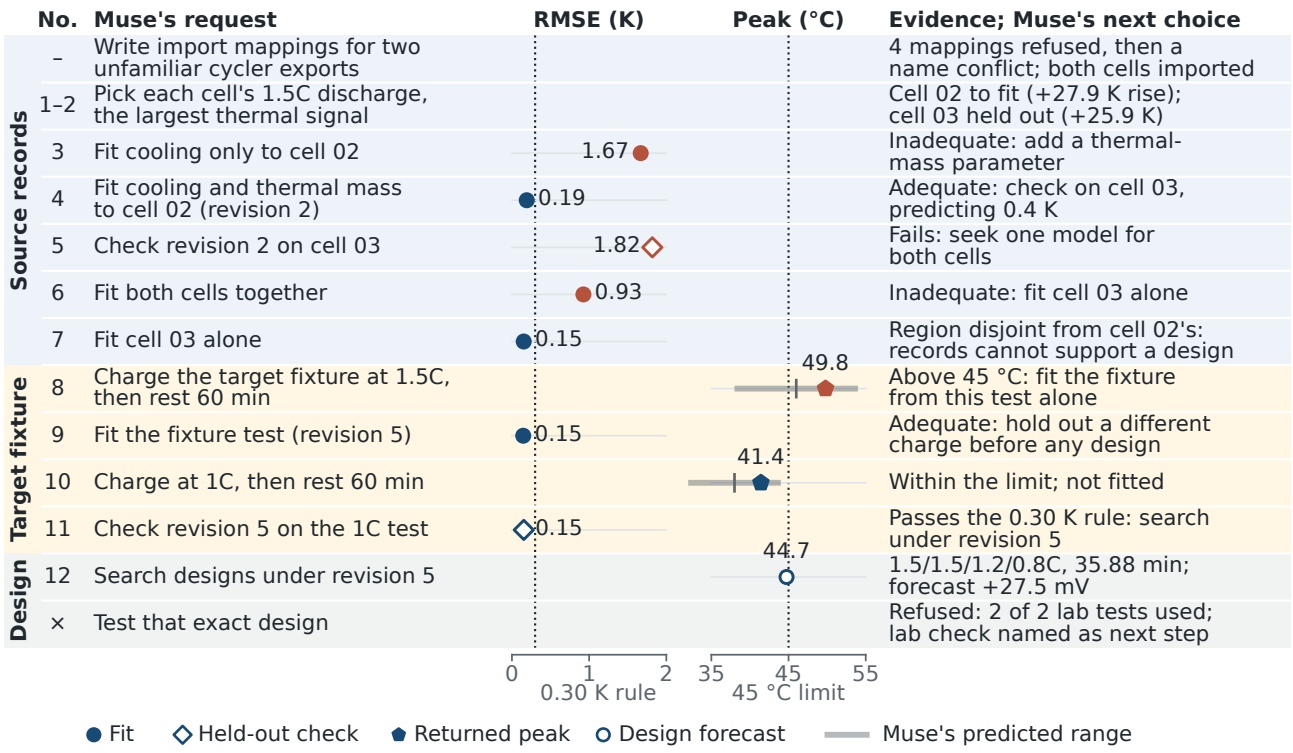


Figure 2: The H1 investigation: each request Muse chose, in execution order from the import to the refused request, with the evidence it returned and the choice it led to. RMSE: fit error (filled) or error on a test not fitted (diamond), against the 0.30 K rule. Peak: returned peak surface temperature of a simulated lab charge, with Muse's prediction (tick) and stated range (bar), or the design's forecast (open circle), against the 45.0°C limit. Blue meets the rule or limit; rust fails it.

Four returned results changed the next action: the failed cooling-only fit (add thermal mass), the failed cross-cell check (pool the cells), the disjoint per-cell fits (stop relying on the records and measure the fixture), and the 1.5C test above the limit (hold out a gentler charge). Numerical routines computed every result; which to run, on which data, and when the evidence sufficed were Muse's choices.

5 Evaluation

We report three charging studies with different questions, arms and budgets, which we keep separate, and two evaluations of evidence checks and intervention decisions that use recorded physical data.

5.1 Records intake: preparation, recommendations and control

Can Translucent investigate raw exports without per-case preparation, and what does enforcement add? H1/H2 supply two cells' exports and withhold another; H3 reformats values with decimal commas, semicolons and ambiguous temperature headers. Every case has the objective of Section 4. Hidden simulations use fits to the withheld cells, with inadequate source RMSEs of 1.30 K (H1), 0.98 K (H2) and 1.07 K (H3). They define in-family scenarios rather than validated physical cells. *Prepared numerical* receives per-case mappings and a sequence: fit source records, check another record, pool, search, test a candidate and refit. It omits the held-out target-fixture check, precluding support and leaving a matched prepared investigation untested.

Translucent recommends in 5/6 runs; the Unenforced workflow in 6/6. Each has five supported recommendations, and all 11 recommendations are feasible in hidden simulation (Figure 4a). Prepared

Case	Translucent	No feedback	Constrained BO	Compass-search reference
A: fixed	33.79	36.29	42.62	33.55
B: amended	37.17	44.29	42.62	37.13
C: extrapolation	34.13	39.50	45.98	34.13

Table 1: Median best feasible charge times (min) after eight simulated experiments, with three runs per model arm and five per Constrained BO case. One No feedback run in C finds no feasible result and ranks as infinitely slow.

numerical recommends in 0/3; its last H3 candidate is also infeasible, at +22.3 mV against the +25.0 mV limit. Imports are exact and complete in 11/12 model runs. No person corrects an import, fit or configuration in any of the 12 model runs or three prepared runs. The Unenforced workflow's unsupported recommendation has 1.12 K fit RMSE against the 0.30 K rule; its interface prevented fitting/checking follow-up tests, and the model acknowledged the lack of support. The Translucent run that declines stops after US$2.15 because another model call could exceed the cap, not because of an evidence judgment. Its mapping dropped five rows per file; a later scorer correction confirmed all retained values matched, and the decline stands.

The Unenforced workflow reaches the same case-specific protocol in 5/6 runs, so what the two arms share already carries much of the preparation. Translucent's enforcement adds control: zero unsupported recommendations among five issued versus one among six, request-bound approvals and an offline-checkable record. Translucent spends US$11.88 versus US$4.63 (2.6 times the Unenforced workflow's model spend) without raising the recommendation rate; mean recorded duration is 950 versus 1,573 s. Translucent spent both lab tests on diagnostic charges, searching only after the held-out check, so none of its five recommended protocols was measured within budget. The Unenforced workflow searched before its first lab test and measured its final recommendation in all six runs. The comparison cannot attribute this order difference to intake instructions, interface or enforcement. Measuring the recommended protocol supplies direct temperature evidence, but does not validate its anode potential.

5.2 Fixed-model charging: better designs under the same experiment allowance

This separate study asks whether experimental feedback improves protocol selection, with one fixed model for search and scoring and no model learning. Each campaign has eight simulated experiments. *Constrained BO* means constrained Bayesian optimization with Gaussian-process surrogates and constrained expected improvement [Gardner et al., 2014]: three Sobol points, then five acquisition steps. Translucent chooses sequentially; in *No feedback* Muse commits eight protocols upfront. Cooling is fixed at 6 $\mathrm{W\,m^{-2}\,K^{-1}}$. Cases A and B use 25.0°C; C uses 15.0°C, outside the parameter set's fitted range and without a registered success criterion. All impose a 45.0°C ceiling. A and C require +10.0 mV; B raises this to +25.0 mV after experiment four. Feasibility additionally requires 0.5 mV and 0.10 K clearance. Final best times use the final constraint; infeasible counts use the constraint when results arrive.

Closing the loop wins here, with more model calls: every Translucent run beats every Constrained BO and No feedback run in its case (Table 1; Figure 3). Translucent's medians are 8.83, 5.45 and 11.85 min faster than Constrained BO's in A, B and C, from three Translucent and five Constrained BO runs per case, with the same eight-experiment allowance. These are better designs under the same allowance, not fewer experiments. The registered A and B criteria are met. One B run reaches 37.03 min, below the Compass-search reference's 37.13 min; all C runs reach its 34.13 min in six or seven simulated experiments. Median infeasible counts are 4/4/4 for Translucent and 2/3/2 for Constrained BO. Recorded model cost is US$1.00–1.25 versus zero, and campaign duration 3.6–5.2

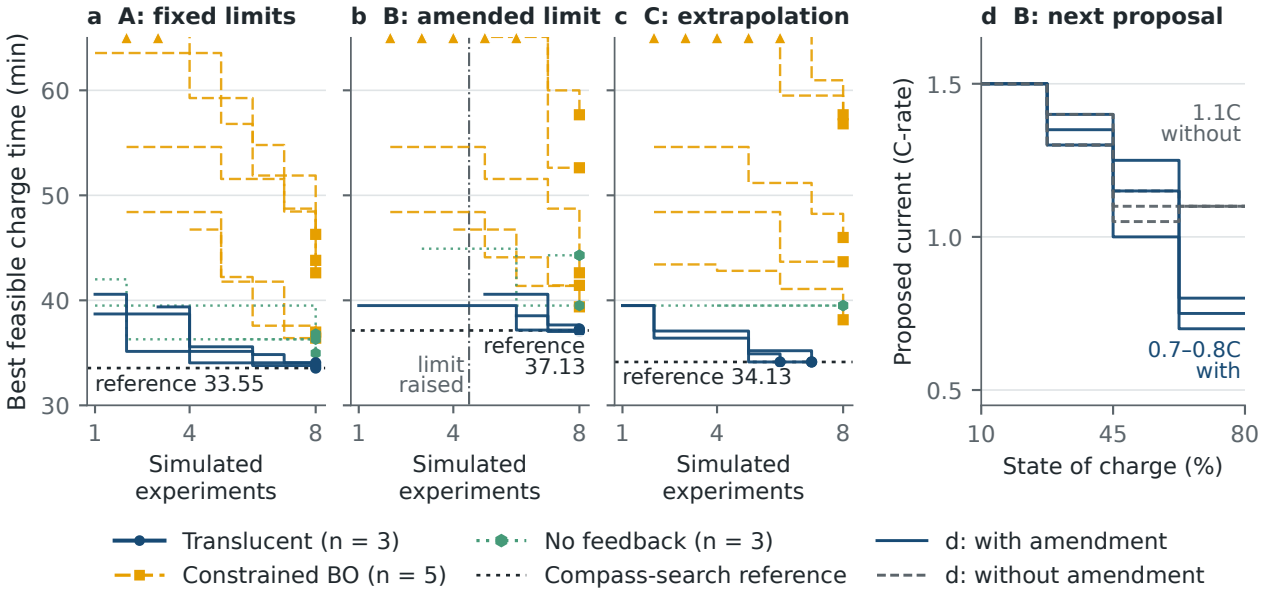


Figure 3: Fixed-model study, eight simulated experiments per campaign. (a–c) Every run's search history. Missing values mean no feasible result; triangles mark finite values above 65 min. B is scored under the final constraint, with its amendment marked. C's Translucent histories stop after 6, 7 and 7 simulated experiments; repeats 1 and 3 reach the next-call spending limit and repeat 2 concludes. (d) Replay from each B run's identical four-experiment prefix: the next proposal with the raised anode limit and without it.

versus 1.0–1.2 min. Fourteen runs truncated by a 4,000-token limit or process termination were set aside and repeated in full, costing US$6.48 separately from the scored runs.

Sequential feedback separates the model arms: in medians, Translucent's loop beats the same model without feedback by 2.50, 7.12 and 5.37 min in A, B and C. In B, where the limit rises mid-campaign, No feedback's median is 1.67 min slower than Constrained BO's and Translucent's 5.45 min faster, although one No feedback run (39.50 min) beats four of five Constrained BO runs. In A and C, No feedback already leads Constrained BO by 6.33 and 6.48 min, which prior charging knowledge plausibly explains. Feedback also changes the number of model calls, so its causal effect is not isolated. Larger-budget BO and language-assisted BO are untested. The loop adapts: a replay from each identical four-experiment B prefix gives final-step currents of 1.1C without the amendment and 0.75/0.7/0.8C with it, forecasting +27.8/+28.0/+25.0 mV (Figure 3d). All three meet the replay's changed-proposal criterion; the last lies exactly on its threshold and lacks the additional race clearance. This tests one next proposal, not a full alternative campaign.

5.3 Model learning: matching a prepared sequence without one, at higher test cost

Can Translucent adapt the investigation when an initial thermal model is wrong? Five conditions test nominal behavior (N), changed cooling (K), noise (U), increased thermal mass (M) and chamber-temperature offset (O). K denotes the source report's cooling case, distinct from fixed-model Case C. Inputs are prepared for both arms. *Adaptive numerical* follows a prepared sequence: search under the initial model, test, fit cooling and thermal mass, assess another measurement's value, then revise and test. Translucent gives Muse the available actions and scientific guidance without this sequence. Both use the same design routine, withhold fixture parameters and assess the final protocol once outside the budget.

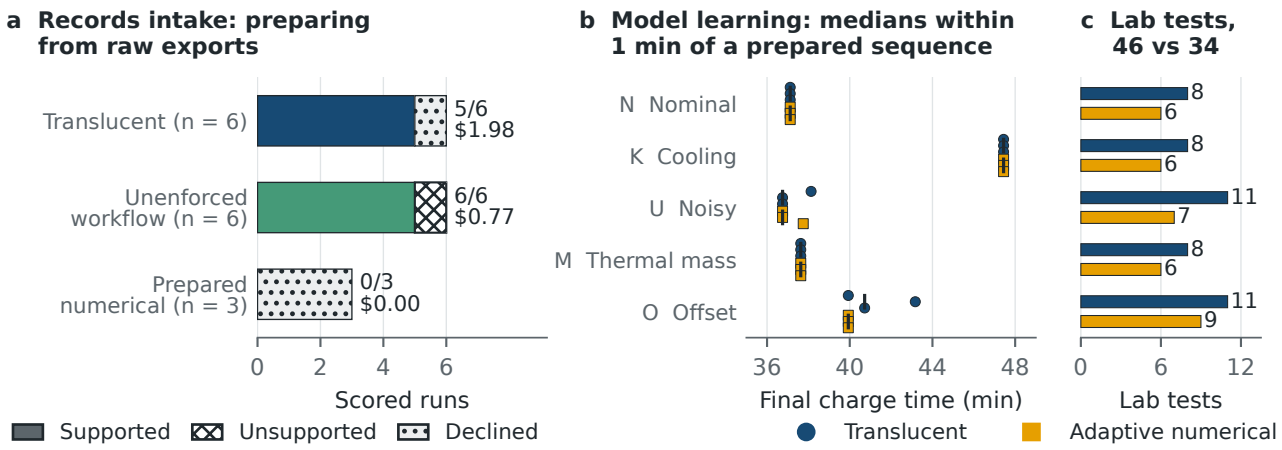


Figure 4: Language-agent investigation and numerical optimization, in two separate studies. (a) Records intake under two lab tests and a US$2.50 model cap: recommendations per arm and mean model spend per run (US$, excluding numerical computation). Every recommendation is feasible in hidden simulation. (b) Model learning: final hidden-condition charge time of each run, three per arm per condition, with median ticks; points are separated vertically only to reveal repeated values, and O uses the amended replacement run. (c) Simulated lab tests per condition, summed over runs.

Translucent and Adaptive numerical each produce 15/15 feasible protocols: 80% state of charge, at least +25.0 mV and at most 45.0°C. Medians match in N/K/U/M at 37.12/47.44/36.75/37.63 min. O takes 40.72 min for Translucent versus 39.93 min, a tie under the descriptive 1.0 min tolerance (Figure 4b). An amendment replaced a run whose measurement-ranking routine exceeded its time limit. That run continued to a 41.18 min design; retaining it would classify Translucent as worse in O. Under its written guidance, Translucent checks held-out observations before its first design search in 13/15 runs; Adaptive numerical does so in 0/15 and checks at all in 4/15. Translucent uses 46 versus 34 lab tests, 143 versus 116 requests and US$16.72 versus zero model spend (Figure 4c). Its noisy-condition stopping criterion, a feasible design with at most one lab test after the first fit, is met in 0/3; one run's extra tests end at 38.13 min, slower than every Adaptive numerical run in U. Offset diagnostics yield more conservative designs without improving on Adaptive numerical.

Two ablations are feasible in 3/15: *Without diagnostics* (no fitting, checking or ranking) always returns the initial model's design, 1.5/1.5/1.05/0.8C, feasible only in N, and *Fixed-model* runs eight simulated experiments under fixed cooling, without a lab. Every condition except N therefore requires model revision; this shows that diagnostics are needed, not that they benefit Muse specifically. *Calibrate once* numerically searches, tests, fits cooling alone, searches again and stops. It yields 12/15 feasible assessments, failing O; in M it takes 44.24 min under an inadequate model versus 37.63 min for Adaptive numerical.

5.4 Evidence checks and intervention decisions

These cases test the multimodal mode's deterministic checks on prerecorded battery microscopy [Reisecker et al., 2023], separately from every charging study: each constructed case introduces a fault and proposes a finding consistent with the faulty measurement. Translucent's checks block 38/38 faults. On 24 shared calibration, timing, recording-identity, segmentation and corroboration cases, they block 24/24 versus 1/24 for a plain pipeline, which detects only crashes or nonfinite output. Fourteen further cases, such as fabricated measurements, have no pipeline analogue. Checks were extended after an earlier suite run, so this is a development-suite result, not held-out validation; some clean outlines are scripted. A separate text-only assessment covers the 18 faults that reach a proposed

finding; 20 stop earlier. With three repetitions and majority acceptance, Muse Spark Open [Meta, 2026b], served by Fireworks in October 2026 without an archived immutable checkpoint, accepts 5/18 faulty findings from numbers and recording labels, versus 0/18 with fuller source relationships and competing evidence; it accepts 1/2 and 2/2 clean controls.

The NIST benchmark uses the 50 interrupted-heating trials with temperature traces, in eight cell-format/state-of-charge conditions: 31 ignite and 19 do not [Tam et al., 2026, National Institute of Standards and Technology, 2025]. Scripted policies stand down, escalate or purchase observation. Assumed costs are 100 for unsafe stand-down or ignition during observation, 10 for unnecessary escalation, 0.05 per observation second and 10 once for abstention. Escalation on an igniting trial costs zero, although suppression efficacy was unmeasured, so always escalating costs $10 \times 19/50 = 3.80$. The primary analysis leaves one condition out, uses a 10 s thermocouple lag and includes inferred heater-off times; outcomes at alternative stopping times were unobserved.

Scripted logistic search costs 3.72 versus 3.80 for threshold search and always escalating, from one fewer unnecessary escalation (18 versus 19) partly offset by observation cost; their 95% condition-block bootstrap intervals, conditional on recorded decisions, overlap, and the other seven scripted policies cost more, though three reach 3.80 when the same cost rule is applied post hoc. Both searches have raw and Holm-adjusted $p = 1$ against always escalating; the primary analysis establishes no advantage. No comparison survives Holm correction in the condition-level sensitivity. In sensitivity analyses, logistic costs rise to 3.83–4.49, with one unsafe stand-down under sequential evaluation, and leaving out a cell format gives threshold search 16.40 with seven unsafe stand-downs. The public data had been explored before protocol amendments. Earlier exploratory Muse campaigns, excluded here, yielded best costs of 3.00 and 3.40 (both Holm-adjusted $p = 1$; the first could see an ignition-correlated flag); direct model decisions made four and six unsafe stand-downs versus zero from model-selected executable procedures, in comparisons chosen after seeing results.

6 Discussion

Investigation versus optimization. The frontier claim is that a language model took part in formulating and carrying out experimental engineering work, not only in optimizing a prepared problem. In H1 it decided that the records could not support a design, which fixture tests could, and when the evidence sufficed (Section 4); in case B it changed its next proposal when the anode limit was raised (Figure 3d). Across both model arms, 11/12 intake runs recommended from raw exports without per-case mappings or a prescribed sequence, ten with passing target-fixture checks and all 11 feasible in hidden simulation. Supported designs were within 0.00–0.24 min of the Hidden-condition reference. Engineer-time savings and historical reliability were unmeasured, and no recommendation was reproduced on hardware.

What Translucent adds. In intake and model learning, Translucent's contribution lay in assembling and checking the investigation around an optimization problem; it did not beat Adaptive numerical's final designs. All five Translucent recommendations and five of six Unenforced recommendations met the support rule, but differences in interfaces and instructions prevent isolating enforcement's causal effect.

What stays with engineers. The autonomy shown is computational: every lab test was simulated, and physical tests, calibration and specimen custody remain with people, who also supply the tools, model families and limits. In a browser run, an approved physical charge was cancelled without an operator and the investigation continued in simulation; a separate scripted-planner test uploaded labelled stand-in data, which Translucent checked and ingested. Both exports passed offline checking, exercising control and ingestion, not physical validation.

Who could benefit. Battery-development teams, and in principle teams developing energy hardware, electronics and other physical systems, repeat the same preparation for each new dataset, fixture or

test plan. A model that carries that preparation through experiments, under approvals and a checkable record, could reduce the repetition; this is a hypothesis, not a measured saving, and a physical-test request is not an executed laboratory experiment.

7 Limitations

In-family intake fixtures ease near-reference recovery after target fitting. Cell disagreement could reflect calibration, specimens or missing physics; its cause is unidentified, and Muse's response shows that tested fits failed declared checks, not that available physics cannot explain the cells. The offset condition combines hidden feasibility with model inadequacy. Nominal regions contain the true parameter in 46/50 simplified-solver fits, with poorer low-cooling coverage. Thermal fitting does not validate physical electrode potential or plating.

Small samples and public records potentially present in training limit inference. Every model-run charging investigation, in Translucent and the Unenforced workflow, used Muse Wonder 1.0; transfer to other models, laboratories and chemistries is untested, as is a prepared intake sequence with target fitting/checking. Adaptive numerical covers every model-learning condition; intake confounds interfaces, and No feedback reduces model calls. The only Translucent decline was a budget stop; no model run declined for inadequate evidence. Microscopy cannot estimate real fault rates. NIST shares fitted models, uses reconstructed events, finite windows and ignition-correlated inferred times.

Recommendation outcomes are frozen. Public code and selected campaigns permit inspection; incomplete archives and the missing frozen source tree prevent public regeneration of every score.

8 Conclusion

Within a supported environment, a language agent did substantial experimental preparation and investigation: from raw battery records, with supplied tools, model families and guidance, Translucent assembled per-case import mappings and fit-and-check sequences, and in H1 it requested a design only after a held-out check on the target fixture passed. What it shares with the Unenforced workflow carried much of this capability; its enforcement kept every recommendation supported and added request-bound approvals and a persistent record, at higher model spend. Translucent's loop beat the same model without feedback by 2.50–7.12 min and Constrained BO at a small fixed-model budget, with more model calls, but did not outperform Adaptive numerical, which matched its designs with fewer tests. The result is a step from optimizing prepared problems toward taking part in the experiment. Before it generalizes to independent laboratories, it needs prospective unfamiliar fixtures and record formats, matched tools and budgets for strong numerical and general-agent comparators, other models, and independent physical measurements of recommended designs under human-approved procedures. The implementation and evaluation reports are publicly available at `https://github.com/ychampion/translucent`.

References

P. M. Attia, A. Grover, N. Jin, K. A. Severson, T. M. Markov, Y.-H. Liao, M. H. Chen, B. Cheong, N. Perkins, Z. Yang, P. K. Herring, M. Aykol, S. J. Harris, R. D. Braatz, S. Ermon, and W. C. Chueh. Closed-loop optimization of fast-charging protocols for batteries with machine learning. *Nature*, 578:397–402, 2020. doi: 10.1038/s41586-020-1994-5.

D. A. Boiko, R. MacKnight, B. Kline, and G. Gomes. Autonomous chemical research with large language models. *Nature*, 624:570–578, 2023. doi: 10.1038/s41586-023-06792-0.

A. M. Bran, S. Cox, O. Schilter, C. Baldassari, A. D. White, and P. Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6:525–535, 2024. doi: 10.1038/s42256-024-00832-8.

C.-H. Chen, F. Brosa Planella, K. O'Regan, D. Gastol, W. D. Widanage, and E. Kendrick. Development of experimental techniques for parameterization of multi-scale lithium-ion battery models. *Journal of The Electrochemical Society*, 167(8):080534, 2020. doi: 10.1149/1945-7111/ab9050.

M. Doyle, T. F. Fuller, and J. Newman. Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell. *Journal of The Electrochemical Society*, 140(6):1526–1533, 1993. doi: 10.1149/1.2221597.

J. Frochte and H. Schmidt. LLM agents as suppliers of physical system models: Calibration and acceptance in Modelica. arXiv preprint arXiv:2610.25019, 2026.

J. R. Gardner, M. J. Kusner, Z. Xu, K. Q. Weinberger, and J. P. Cunningham. Bayesian optimization with inequality constraints. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 937–945. PMLR, 2014. URL `https://proceedings.mlr.press/v32/gardner14.html`. Part 2.

A. E. Ghareeb, B. Chang, L. Mitchener, A. Yiu, C. J. Szostkiewicz, D. Shved, G. J. Gyimesi, J. M. Laurent, S. M. Wright, M. T. Razzak, A. D. White, S. C. Finnemann, M. M. Hinks, and S. G. Rodriques. A multi-agent system for automating scientific discovery. *Nature*, 655:497–505, 2026. doi: 10.1038/s41586-026-10652-y.

R. T. Lange, L. Grillotti, Y. Yamada, R. Charakorn, and D. Ha. The agentic scientist: Orchestrating coding agents for automated scientific (co-)discovery. arXiv preprint arXiv:2610.18933, 2026.

G. Lei, F. Brosa Planella, S. G. Baird, and S. J. Cooper. From prompt to protocol: Fast charging batteries with large language models. arXiv preprint arXiv:2601.09626v1, 2026.

T. Liu, N. Astorga, N. Seedat, and M. van der Schaar. Large language models to enhance Bayesian optimization. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=OOxotBmGol`.

I. Mandal, J. Soni, M. Zaki, M. M. Smedskjaer, K. Wondraczek, L. Wondraczek, N. N. Gosvami, and N. M. A. Krishnan. Evaluating large language model agents for automation of atomic force microscopy. *Nature Communications*, 16:9104, 2025. doi: 10.1038/s41467-025-64105-7.

Meta. Models: Muse Wonder. Meta Model API documentation, 2026a. URL `https://dev.meta.ai/docs/models`. Accessed 4 November 2026.

Meta. Muse Spark Open: Run inference. Meta Model API documentation, 2026b. URL `https://dev.meta.ai/docs/muse-spark-open/deploy`. Fireworks provider model identifier: accounts/fireworks/models/muse-spark-open. Accessed 4 November 2026.

K. Murphy. Model discovery agent: LLM-assisted Bayesian experiment design for data-efficient discovery of mechanistic world models. arXiv preprint arXiv:2608.09696, 2026. Version 5, revised 27 September 2026.

National Institute of Standards and Technology. Determination of safe intervention windows for 18650 and 21700 lithium-ion batteries from external heating: Thermocouple data. NIST Public Data Repository, ark:/88434/mds2-4027, version 1.0.0, 2025. NIST open licence.

S. E. J. O'Kane, W. Ai, G. Madabattula, D. Alonso-Alvarez, R. Timms, V. Sulzer, J. S. Edge, B. Wu, G. J. Offer, and M. Marinescu. Lithium-ion battery degradation: how to model it. *Physical Chemistry Chemical Physics*, 24(13): 7909–7922, 2022. doi: 10.1039/D2CP00417H.

B. Planden, N. E. Courtier, M. Robinson, A. Khetarpal, F. Brosa Planella, and D. A. Howey. PyBOP: A python package for battery model optimisation and parameterisation. *Journal of Open Source Software*, 10(116):7874, 2025. doi: 10.21105/joss.07874.

B. Ranković, R.-R. Griffiths, and P. Schwaller. Large language models as uncertainty-calibrated optimizers for experimental discovery. *Nature Machine Intelligence*, 8:1466–1477, 2026. doi: 10.1038/s42256-026-01283-z.

V. Reisecker, F. Flatscher, L. Porz, C. Fincher, J. Todt, I. Hanghofer, V. Hennige, M. Linares-Moreau, P. Falcaro, S. Ganschow, S. Wenner, Y.-M. Chiang, J. Keckes, J. Fleig, and D. Rettenwander. Effect of pulse-current-based protocols on the lithium dendrite formation and evolution in all-solid-state batteries. *Nature Communications*, 14: 2432, 2023. doi: 10.1038/s41467-023-37476-y.

Y. Ruan, C. Lu, N. Xu, Y. He, Y. Chen, J. Zhang, J. Xuan, J. Pan, Q. Fang, H. Gao, X. Shen, N. Ye, Q. Zhang, and Y. Mo. An automatic end-to-end chemical synthesis development platform powered by large language models. *Nature Communications*, 15:10160, 2024. doi: 10.1038/s41467-024-54457-x.

V. Sulzer, S. G. Marquis, R. Timms, M. Robinson, and S. J. Chapman. Python battery mathematical modelling (PyBaMM). *Journal of Open Research Software*, 9(1):14, 2021. doi: 10.5334/jors.309.

W. C. Tam, H. Fang, J. Chen, Q. Wang, C. Brown, M. I. S. Emon, and A. Putorti. Examining safe intervention windows for 18650 and 21700 lithium-ion batteries under heat interruption experiments. *Applied Thermal Engineering*, 298:131107, 2026. doi: 10.1016/j.applthermaleng.2026.131107.

Y. Wen and H. Liu. Autonomous science bench: Evaluating AI agents for autonomous scientific discovery. arXiv preprint arXiv:2610.18449v1, 2026.

L. Yao, S. Samantray, A. Ghosh, K. Roccapriore, L. Kovarik, S. Allec, and M. Ziatdinov. Operationalizing serendipity: Multi-agent AI workflows for enhanced materials characterization with theory-in-the-loop. arXiv preprint arXiv:2508.06569, 2025.