BKBAGENT: A BATTERY PARAMETER IDENTIFICATION AGENT WITH BAYESIAN EXPERIMENTAL DESIGN AND KNOWLEDGE GRAPH

Anonymous authors
Paper under double-blind review

ABSTRACT

Electrochemical battery models support battery design, state estimation, and control, but their accuracy depends on identifying internal parameters from operating measurements such as voltage. Existing work has applied large language model (LLM) agents to this task, using physical reasoning and simulation feedback to guide parameter updates. However, identifying coupled battery parameters from limited measurements is ill-posed: different parameter combinations can produce nearly identical responses. We introduce BKBAgent, a battery parameter identification agent that combines a physics-informed knowledge graph with Bayesian experimental design (BED). The agent maintains a joint belief over plausible parameter combinations and uses retrieved physical knowledge to propose candidates and anticipate their responses. Using these predictions, BED selects the next candidate by its expected change in the belief over parameter hypotheses. Simulator feedback updates the belief, and the agent retains parameter combinations whose simulated responses match the measurements. We evaluate BKBAgent on a benchmark comprising 108 test tasks with multi-parameter perturbations and large single-parameter shifts across operating rates. Under the same simulation budget, it reduces mean parameter identification error by 25.1% and 23.8% on the two task types, respectively, compared with an LLM-agent baseline. Ablations show that removing either retrieved physical knowledge or BED increases parameter identification error. These findings illustrate how structured domain knowledge and information-guided simulation can support scientific agents in reasoning over competing hypotheses and updating beliefs from feedback.

1 INTRODUCTION

Electrochemical battery models provide a physical basis for battery design, state estimation, and control (Chen et al., 2020; Sulzer et al., 2021). Applying these models to a particular cell requires identifying its material, geometric, and kinetic parameters (Doyle et al., 1993). Simulation-based identification typically proceeds through repeated cycles of proposing parameter values, simulating the recorded protocols, comparing the resulting responses with the measurements, and refining the parameters (Forman et al., 2012; Chen et al., 2026). A candidate may reproduce the discharge voltage reasonably well while still failing to match the current decay during constant-voltage charging, requiring further adjustment and verification. Each evaluation solves coupled electrochemical equations, making repeated searches over a large, continuous parameter space expensive.

Evolutionary and Bayesian optimization guide this process by using curve-fitting error to select successive candidates (Forman et al., 2012; Hansen et al., 2019; Snoek et al., 2012). An aggregate error measures agreement with the data but provides limited guidance about the physical causes of the remaining discrepancies. Large language model (LLM) agents can incorporate physical knowledge and detailed simulation feedback into this iterative process. Battery-Sim-Agent (BSA), for example, interprets response mismatches using domain knowledge and execution memory to propose targeted parameter corrections (Chen et al., 2026).

However, repeated improvements in curve fitting do not by themselves ensure accurate parameter recovery. As illustrated in Figure 1, ohmic loss, charge transfer, and concentration polarization jointly

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.

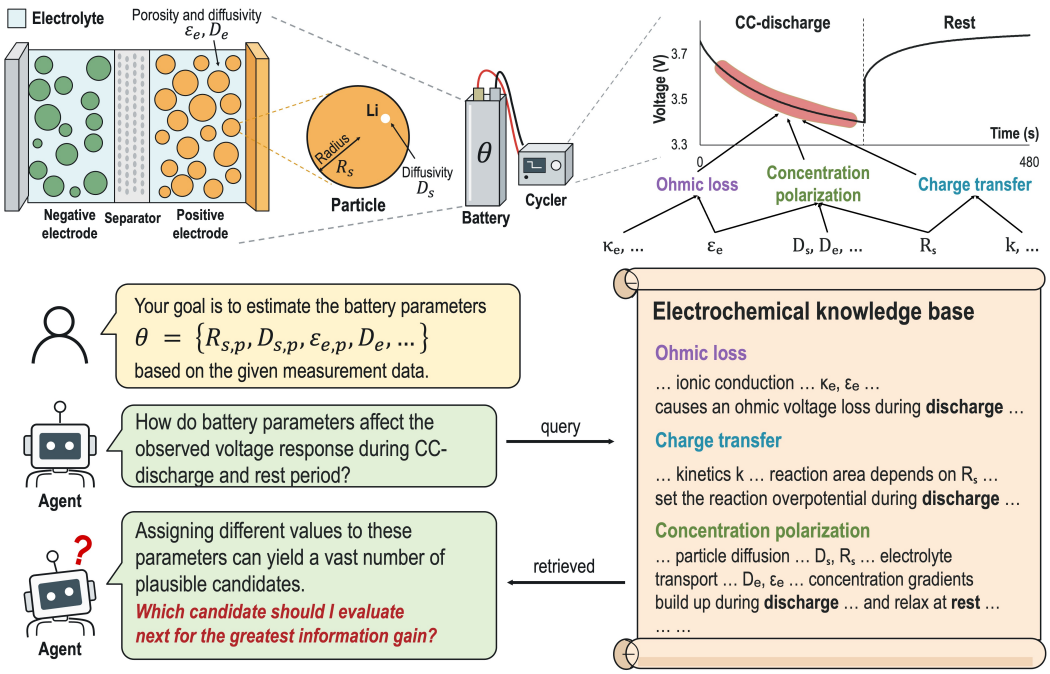

Figure 1: Illustration of agent-based battery parameter identification and its key challenge.

shape the discharge-voltage trajectory. Changes in their associated parameters can compensate for one another, allowing different parameter combinations to produce similarly accurate fits (Bizeray et al., 2019; Aitio et al., 2020). BSA's correction loop does not explicitly maintain uncertainty over these alternatives or compare candidate simulations by their anticipated effect on that uncertainty. The remaining challenge is therefore to account for plausible parameter combinations when deciding which simulation to run next.

We address this decision by linking physical knowledge to a joint belief over parameter vectors. A knowledge graph (KG) connects observable features to processes, equations, parameters, and applicable conditions, providing a basis for reasoning about coordinated parameter changes. The belief assigns a weight to each complete vector, representing alternative explanations of the measurements while preserving parameter dependencies. Bayesian experimental design (BED) then compares candidate simulations by their expected effect on this uncertainty (Rainforth et al., 2024). Selected vectors are simulated under the recorded protocols and compared with the measured responses (Oliveira et al., 2024; Reichelt et al., 2026).

We introduce BKBAgent, which combines physics-informed KG retrieval with BED for battery parameter identification. The agent retrieves physical relations using measured response features and protocol context, then uses them to initialize its joint belief. At each iteration, the LLM uses this knowledge, belief, and execution history to propose parameter vectors together with their possible simulation outcomes. Each outcome specifies response features and an associated probability. Following prior LLM-based information-gathering methods (Choudhury et al., 2026; Ruhrberg Estévez et al., 2025), the LLM estimates how each hypothetical outcome, compared with the measurements, would change the belief. BED selects the candidate with the largest predicted belief-change utility; actual simulator feedback then updates the current belief and execution history. The agent returns a finite set of complete vectors that pass common response-matching criteria.

Our reported benchmark contains 108 identification tasks based on the Doyle–Fuller–Newman (DFN) electrochemical model (Doyle et al., 1993), spanning two material parameterizations and three operating rates. Following BSA (Chen et al., 2026), Regular tasks perturb several parameters, while Extreme tasks apply a large perturbation to one. Under the same simulation budget, BKBAgent reduces mean normalized parameter RMSE relative to BSA by 25.1% and 23.8%, respectively. Matched ablations show that removing physical knowledge or BED increases identification

error, and KG retrieval outperforms text retrieval from the same sources. Further tests cover outcome predictions, belief updates, three LLM backbones, and frozen-point prediction under unseen protocols and on one measured cell.

2 Related Work

2.1 Battery Parameter Identification

Battery model parameters can be determined through dedicated experiments or by fitting simulated responses to measurements (Chen et al., 2020; Forman et al., 2012). Black-box optimization supports the latter approach when evaluations are expensive or gradients are unavailable (Snoek et al., 2012; Hansen et al., 2019). Efficient optimization, however, does not ensure identifiable parameters: several vectors may produce equally good fits. Structural identifiability analysis, profile likelihood, and Bayesian inference characterize which parameters the observations can constrain (Bizeray et al., 2019; Raue et al., 2009; Aitio et al., 2020). More broadly, learned posterior estimators represent joint parameter uncertainty in simulation-based inference (Greenberg et al., 2019; Radev et al., 2022).

Scientific agents guide the search by relating response patterns to physical processes. BSA uses a knowledge warm-up stage, multimodal feedback, and dynamic memory to make physically motivated parameter updates (Chen et al., 2026). Building on this use of domain knowledge, we maintain a belief over multiple complete vectors and compare candidate simulations by their expected information gain. This adds an explicit uncertainty-based criterion to parameter correction.

Experimental design links inference to the choice of observations or computations. PIED uses physical structure to guide data collection for inverse problems (Hemachandra et al., 2025). BACON and climate-model calibration select informative simulations using existing observations (Oliveira et al., 2024; Reichelt et al., 2026). We adopt this computational form of BED: the measurements and protocols are given, and the agent chooses which parameter vector to simulate next. Knowledge retrieval and LLM reasoning guide the predictions and belief assessments used in this decision.

2.2 LLMs for Scientific Discovery

LLM agents connect knowledge-based proposals to external evaluation. Reasoning-and-action frameworks link textual decisions to tools, while experience-based agents use feedback to refine later decisions (Yao et al., 2023; Shinn et al., 2023). LLM-SR applies this pattern to equation discovery through candidate generation, numerical evaluation, and an experience buffer (Shojaee et al., 2025). LLAMBO uses language models for candidate sampling and surrogate prediction in expensive optimization (Liu et al., 2024). These methods motivate joint candidate generation and outcome prediction, with a simulator providing the feedback for subsequent reasoning.

Knowledge-graph retrieval provides connected relations as context for LLM reasoning (Hu et al., 2025; He et al., 2024; Li et al., 2025). In battery identification, these relations link response features to electrochemical processes, governing equations, and parameters under specified conditions. We retrieve applicable subgraphs at initialization and reuse them when generating candidates, predicting outcomes, and interpreting simulator feedback. The KG supplies physical relations; a separate distribution over complete parameter vectors represents the agent's belief.

Choosing an action also requires assessing what its outcome could reveal. BED-LLM selects queries using probabilistic predictions, and ACTMED compares predicted test outcomes for sequential diagnosis (Choudhury et al., 2026; Ruhrberg Estévez et al., 2025). Both use anticipated belief changes to guide information gathering. In battery identification, the simulator input is chosen by the agent, whereas the target parameters belong to the measured cell. We therefore assess each predicted outcome against the fixed measurements before estimating its effect on the parameter belief.

3 Methodology

BKBAgent uses physical knowledge to initialize a joint parameter belief, proposes candidates with their possible outcomes, and uses BED to select the next simulation (Figure 2). Actual simulator feedback updates the belief and guides the next iteration.

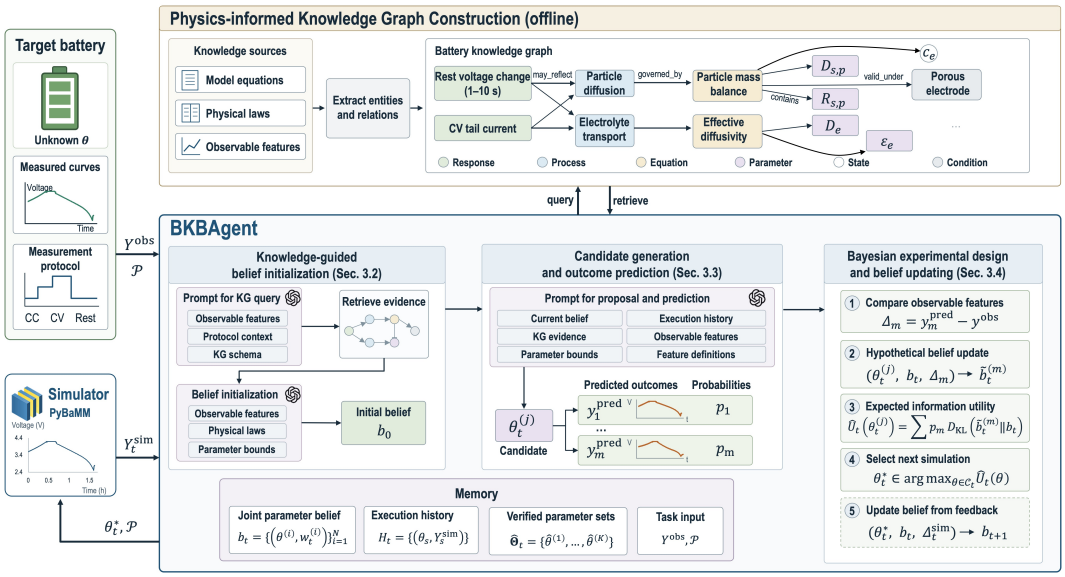

Figure 2: Illustration of the BKBAgent framework for battery parameter identification. Utility denotes the approximate expected belief change in Eq. (3).

3.1 Problem Formulation

The inputs are battery measurements Y^{obs} and their protocols $\mathcal{P}$. For each protocol p, let Y_p^{obs} and $Y_p^{\text{sim}}(\theta)$ denote the measured and simulated responses, where $\theta \in \Theta$ is a parameter vector within the model's physical bounds. Response matching (Chen et al., 2026) measures agreement as

$$\mathcal{L}(\theta) = \sum_{p \in \mathcal{P}} w_p \, d\big(Y_p^{\text{sim}}(\theta), Y_p^{\text{obs}}\big) \,, \tag{1}$$

where d measures response discrepancy and $w_p \geq 0$ are protocol weights. A point estimator minimizes $\mathcal{L}$. We return a finite set $\widehat{\Theta} = \{\widehat{\theta}^{(1)}, \ldots, \widehat{\theta}^{(K)}\}$ of complete vectors whose simulated responses satisfy common calibration criteria. This set need not contain every compatible vector. Given the measurements and protocols, the agent selects vectors to simulate within a fixed budget.

3.2 Knowledge-Guided Belief Initialization

An observable feature may reflect several electrochemical processes, whose contributions depend on the measurement protocol. We extract features y^{obs} for each protocol stage, including voltage slopes, cutoff capacity, constant-voltage (CV) current decay, and relaxation, together with their units, time windows, and operating context. Uppercase Y denotes complete responses; lowercase y denotes extracted features. Given these features, protocol context, and the KG schema, the LLM formulates a physical query and identifies relevant response nodes.

The KG is constructed offline from model equations, physical laws, and observable definitions. Typed relations connect responses, processes, equations, parameters, states, and conditions. Retrieval follows response–process–equation paths, returning connected subgraphs with their quantities, sources, and assumptions. A relation is included only when its model, material, and operating conditions are satisfied. Material parameterizations share physical relations, while their constitutive functions retain material-specific conditions. Retrieved evidence $\mathcal{K}$ is cached per run.

Using $\mathcal{K}$, the measurements, protocols, and parameter bounds, the LLM initializes a finite joint belief. At iteration t, it is represented as

$$b_t = \{(\theta^{(i)}, w_t^{(i)})\}_{i=1}^{N}, \qquad w_t^{(i)} > 0, \quad \sum_i w_t^{(i)} = 1. \tag{2}$$

Each weight expresses the relative plausibility of a complete parameter vector, preserving dependencies among its coordinates (Aitio et al., 2020). For example, a transport relation may suggest that

conductivity and porosity should vary together. The LLM considers such joint changes alongside the other retrieved relations and the measured responses when assigning initial weights.

The belief's hypothesis support stays fixed within a run so that candidate simulations can be compared through weight changes over the same vectors. This support approximates the parameter domain; candidate evaluations and verified outputs may lie outside it. Memory stores b_t, the execution history $H_t = \{(\theta_s, Y_s^{\mathrm{sim}})\}_{s<t}$, the verified set $\widehat{\Theta}_t$, and the task input. Each history record contains an evaluated vector, its simulated response, discrepancies, and solver status.

3.3 Candidate Generation and Outcome Prediction

The LLM jointly proposes parameter vectors and predicts their possible simulation outcomes. The proposal-and-prediction prompt supplies b_t, H_t, $\mathcal{K}$, measured features, parameter bounds, and feature definitions. A single call returns candidates $\mathcal{C}_t = \{\theta_t^{(j)}\}_{j=1}^{J_t}$, each paired with outcomes $\{(y_m^{\mathrm{pred}}, p_m)\}_{m=1}^{M}$, where $p_m \geq 0$ and $\sum_{m=1}^{M} p_m = 1$. Here j indexes candidates and i indexes belief hypotheses. Outcomes, probabilities, and the hypothetical beliefs below are defined separately for each candidate; their dependence on j is implicit.

Each outcome describes a joint set of observable features, using the same definitions, units, and time windows as the measurements. Linked quantities, such as duration and passed charge, are predicted together. The curves in Figure 2 illustrate these predictions schematically; the LLM outputs feature descriptions, not pointwise trajectories. The probabilities concern unexecuted computations conditional on successful execution, not randomness in the deterministic simulator.

The prompt asks how coordinated parameter changes affect responses under the retrieved relations and which parameter hypotheses the resulting comparisons could distinguish. Proposals must satisfy physical bounds, avoid previously attempted vectors, and cover the required features. The resulting candidate–outcome pairs pass to BED without changing b_t or H_t.

3.4 Bayesian Experimental Design and Belief Updating

BED compares candidates by their expected effect on parameter uncertainty (Rainforth et al., 2024). For each candidate, we first compare its predicted outcomes with the measured features: $\Delta_m = y_m^{\mathrm{pred}} - y^{\mathrm{obs}}$. The candidate is a chosen simulator input, whereas the target parameters describe the measured cell. Its outcome therefore informs identification through agreement with the measurements, interpreted using the execution history and physical relations (Oliveira et al., 2024; Reichelt et al., 2026).

Following LLM-based information gathering (Choudhury et al., 2026; Ruhrberg Estévez et al., 2025), the LLM assesses each hypothetical outcome as $(\theta_t^{(j)}, b_t, \Delta_m) \rightarrow \widetilde{b}_t^{(m)}$, with H_t and $\mathcal{K}$ as context. The temporary belief gives the weights that the same hypotheses would receive if this outcome were observed. Each assessment starts from b_t; temporary beliefs neither overwrite it nor accumulate across outcomes.

The evaluator normalizes the supplied probabilities and computes

$$\widehat{U}_t(\theta_t^{(j)}) = \sum_{m=1}^{M} p_m D_{\mathrm{KL}}\left(\widetilde{b}_t^{(m)} \| b_t\right), \qquad \theta_t^{\star} \in \arg\max_{\theta \in \mathcal{C}_t} \widehat{U}_t(\theta). \tag{3}$$

KL divergence is computed over the common hypothesis support. BED selects the candidate with the largest probability-weighted divergence, using the distributions supplied by the LLM. We interpret this as an approximate expected belief-change utility: it includes both separation between outcome-conditioned beliefs and displacement of their average from the current belief. Appendix E separates these terms and evaluates their effect on candidate selection.

The selected vector $\theta_t^{\star}$ is simulated in PyBaMM under $\mathcal{P}$, producing Y_t^{sim} and feature discrepancies $\Delta_t^{\mathrm{sim}} = y_t^{\mathrm{sim}} - y^{\mathrm{obs}}$. The LLM then updates $(\theta_t^{\star}, b_t, \Delta_t^{\mathrm{sim}}) \rightarrow b_{t+1}$ using actual execution records, measurements, and $\mathcal{K}$. Predicted outcomes and temporary beliefs are excluded from this update. Memory adds $\theta_t = \theta_t^{\star}$ and its actual response to H_{t+1}. Cached results do not trigger repeated updates; failed solves consume budget but leave the belief unchanged.

Vectors enter the archive only after simulation on every record and successful full-response validation. A fixed discrepancy-and-distance rule selects at most K complete outputs without access to ground-truth parameters. The run ends at the simulation budget, a round limit, or a recorded failure, as specified in Algorithm 1.

LLM-predicted and conditional distributions need not form a coherent Bayesian joint model, so $\widehat{U}_t$ does not have an exact expected-information-gain interpretation. We check whether probability-weighted conditional beliefs recover b_t (Appendix A) and evaluate prediction accuracy, feedback updates, and selection quality empirically. Simulation can improve understanding of the forward map, but cannot eliminate ambiguity inherent in the fixed measurements.

4 EXPERIMENTS

We evaluate how physical knowledge and BED affect parameter recovery, response matching, and simulation selection. All methods use the same existing measurements; the agent selects computations without acquiring new physical data.

4.1 EXPERIMENTAL SETUP

Tasks and knowledge. The primary synthetic benchmark uses the isothermal DFN model in PyBaMM 25.6.0 (Doyle et al., 1993; Sulzer et al., 2021), with Chen2020 and ORegan2022 parameterizations of the LG M50 cell (Chen et al., 2020; O'Regan et al., 2022). Both describe NMC811/graphite–SiO_x electrodes. Ten unknown parameters span diffusion, transport, capacity, and reaction strength. All methods share parameter bounds, initial conditions, and constitutive-function multipliers. Following BSA (Chen et al., 2026), Regular tasks perturb multiple parameters, whereas Extreme tasks substantially perturb one. Each independent perturbation instance, termed a parent, is paired across both parameterizations and 0.2/1/2C. Development contains 48 tasks. Reported tests contain 108 tasks from ten Regular and eight Extreme parents, after excluding two construction-induced development/test overlaps (Appendix J.2). The KG links observable responses to applicable physical relations through 126 nodes and 341 edges, excluding fitted test cases. Appendix B lists sources and conditions of use.

Measurements and computation. Each synthetic record contains two continuous constant-current/constant-voltage (CCCV) cycles, starting from state of charge (SOC) 0.6 at 298.15 K. We score only the second cycle using CC/rest voltage, CV current, and event durations. Each run allows 100 attempted record solves, including two common initializations, failures, and verification. We compare against BO, CMA-ES, BSA, raw-coordinate computational BED (C-BED), and the GP-based knowledge/BED reference (GP-BKB) (Snoek et al., 2012; Hansen et al., 2019; Chen et al., 2026). BO and BSA use 20 solves for initialization. BSA and BKBAgent share GLM-4.6V. BKBAgent uses 16 joint hypotheses, four candidates per round, three predicted outcomes per candidate, and at most 16 observable features. All methods use three paired search seeds.

Metrics. For parameter identification, the reported point estimate minimizes calibration error without access to ground-truth parameters. We measure parameter recovery by RMSE normalized to the physical parameter ranges (NRMSE). Validated-set success requires at least one vector to pass response validation, while Hit@5 requires one of at most five complete vectors to recover every coordinate within its specified tolerance. All archives use the same acceptance and separation rules. Confidence intervals bootstrap independent parents after averaging seeds and paired tasks. Appendix C defines the scoring rules.

4.2 PARAMETER RECOVERY AND PHYSICAL RESPONSE MATCHING

Table 1 compares point and set-valued identification. BKBAgent achieves the lowest mean parameter NRMSE in both modes: 0.099 on Regular tasks and 0.120 on Extreme tasks, reducing BSA's errors by 25.1% and 23.8%, respectively. Validated-set success reaches 90.6% and 93.8%. Despite these improvements over the baselines, Extreme tasks retain a wider error distribution (Figure 3), indicating that large perturbations remain difficult under the shared budget.

Table 1: Parameter identification under a common simulation budget. Brackets give 95% parent-bootstrap intervals. Valid denotes a nonempty accepted archive; set size includes empty archives. Hit@5 evaluates complete vectors.

Mode	Method	Parameter NRMSE ↓	Valid (%)	Set size	Hit@5 (%)
Regular	BO	0.161 [0.151, 0.170]	24.4	0.39	13.3
	CMA-ES	0.185 [0.175, 0.194]	15.0	0.27	3.9
	BSA	0.133 [0.125, 0.140]	56.7	1.10	25.0
	C-BED	0.136 [0.128, 0.145]	47.8	1.10	22.8
	GP-BKB	0.106 [0.101, 0.112]	90.6	2.19	57.8
	BKBAgent	**0.099** [0.094, 0.105]	90.6	3.19	58.9
Extreme	BO	0.186 [0.169, 0.202]	29.9	0.39	9.0
	CMA-ES	0.207 [0.186, 0.225]	19.4	0.31	7.6
	BSA	0.158 [0.142, 0.172]	66.0	1.35	27.8
	C-BED	0.154 [0.139, 0.166]	57.6	1.15	27.1
	GP-BKB	0.126 [0.113, 0.136]	93.1	2.06	47.2
	BKBAgent	**0.120** [0.107, 0.133]	93.8	3.16	42.4

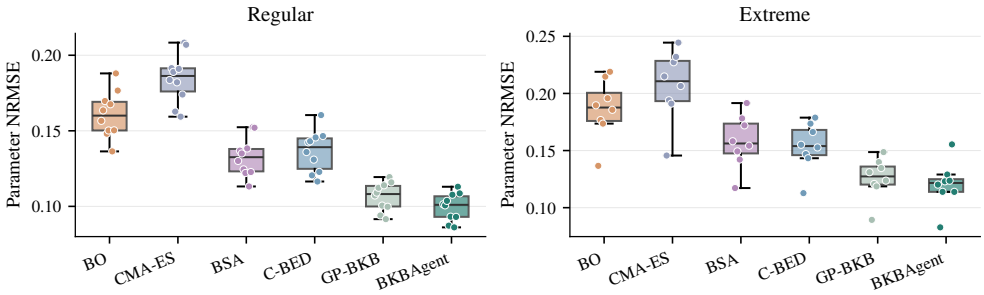

Figure 3: Parent-level parameter errors, averaged over paired parameterizations, rates, and seeds. Boxes show medians and interquartile ranges, complementing the means in Table 1.

Against the closest GP-BKB reference, the mean NRMSE differences are -0.0069 on Regular and -0.0056 on Extreme tasks, with paired 95% intervals $[-0.0075, -0.0063]$ and $[-0.0097, -0.0010]$, respectively. BKBAgent improves on 10/10 Regular and 6/8 Extreme parents (Appendix C). These gains concern the shared simulation budget; API workload is reported separately in Appendix G.

BKBAgent leads in 10 of the 12 operating strata. C-BED leads on Regular ORegan2022/0.2C, and BSA on Extreme ORegan2022/2C (Table 11), showing that the advantage varies with operating conditions. Set-valued recovery also differs from point accuracy: Extreme Hit@5 is 42.4%, below GP-BKB's 47.2%, despite a lower point error and a larger archive. Retaining more response-compatible vectors therefore does not necessarily improve complete-vector recovery.

Figure 4 illustrates validation across voltage, CV current, and transition timing. Appendix G reports parameter recovery and validated-set discovery across simulation budgets.

4.3 Joint Beliefs and Alternative Parameter Explanations

To examine how feedback changes the joint belief, we use four conditional tasks covering diffusion, transport, capacity, and reaction. Each varies two parameters while fixing the other eight. Independent simulator-based reference distributions provide a common comparison target (Lueckmann et al., 2021). Figure 5 shows weighted hypotheses alongside these references and the complete alternative vectors accepted by response validation.

In all three displayed cases, feedback redistributes weight over the fixed support and reduces the marginal cumulative distribution function (CDF) distance to the conditional reference. A joint-CDF comparison also favors the updated beliefs over independent products with identical weighted marginals on all four tasks (Appendix F). Distinct parameter combinations pass response validation,

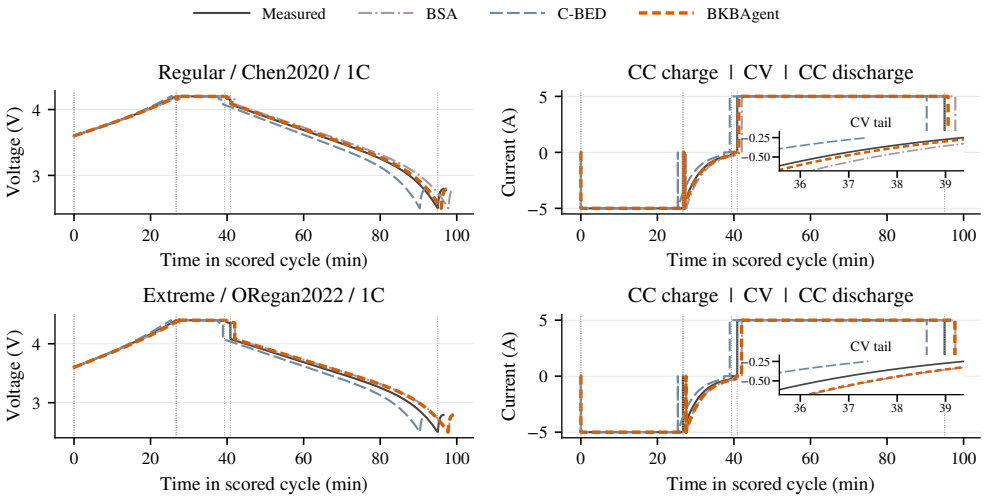

Figure 4: Second-cycle responses for fixed 1C cases: Regular Chen2020 and Extreme ORegan2022, parent 00. Dotted lines mark measured phase boundaries; insets enlarge the CV tail. Physical time is retained to show errors in transition timing.

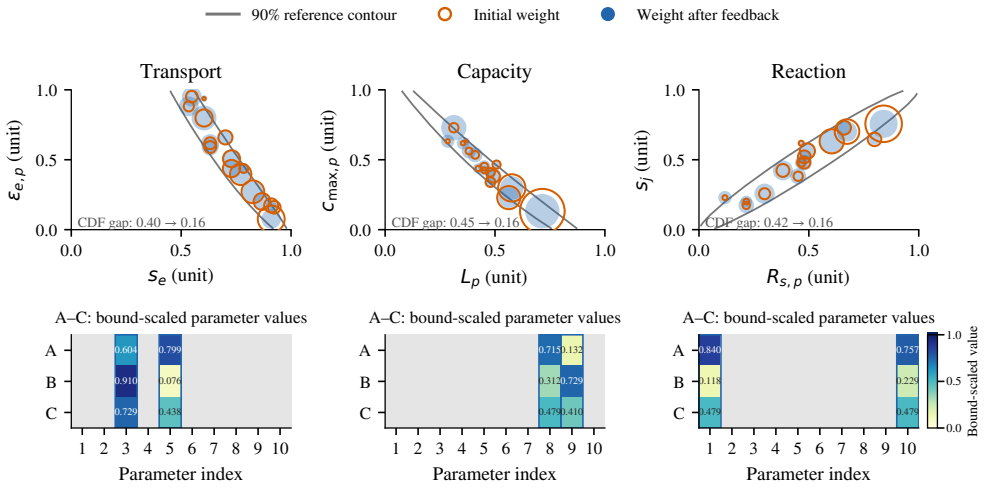

Figure 5: Joint beliefs and complete vectors. Each marker is a full hypothesis projected onto two free coordinates; area gives weight. Orange hollow and blue filled circles use the same area scale for initial and updated weights. Gray curves enclose 90% reference mass. Rows A–C show bound-scaled parameter values: two free coordinates are colored; eight fixed coordinates are gray.

illustrating why a good fit need not identify a unique vector. Archive acceptance depends on actual responses independently of belief weight. These conditional two-parameter cases do not establish calibration of the full ten-dimensional belief.

4.4 Contributions of Physical Knowledge and BED

On 16 paired 1C tasks with three seeds, we compare Full, No external knowledge, Text-RAG, and w/o BED. Text-RAG replaces relational with text retrieval, preserving sources and context budget. W/o BED retains KG evidence, belief, and joint candidate–outcome generation but replaces expected-KL selection with LLM choice. All variants retain actual-feedback updating and common output validation.

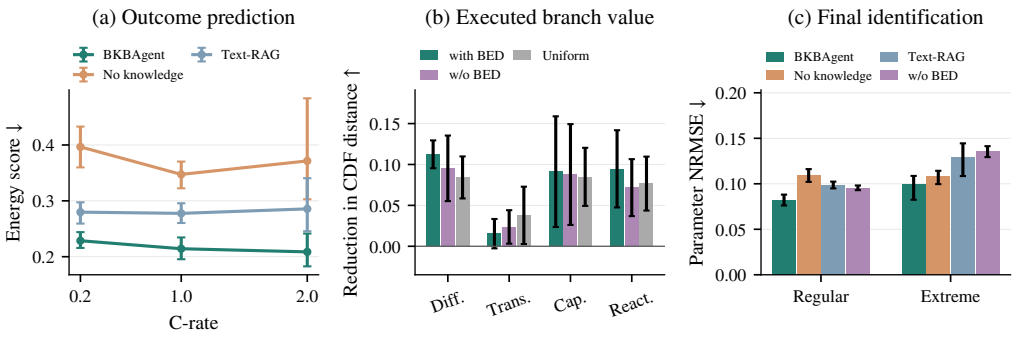

Figure 6: Prediction, selection, and identification. (a) Energy score. (b) Mean marginal-CDF improvement over seeds and snapshots, with seed standard deviations. (c) Matched ablations with parent-bootstrap intervals.

With common candidates, history, and belief, BKBAgent's energy score is 0.217, versus 0.281 for Text-RAG and 0.372 without external knowledge (Figure 6(a)) (Gneiting & Raftery, 2007).

BED yields the largest mean marginal-CDF improvement in three tasks. Transport instead favors uniform selection (0.038) over w/o BED (0.024) and BED (0.015): expected change need not improve accuracy (Figure 6(b)). Appendix E compares five selectors using joint CDF.

Removing knowledge or BED raises identification error; Text-RAG also trails relational retrieval (Figure 6(c)). Appendix H evaluates backbones.

4.5 Real-World Validation and Cross-Protocol Prediction

CALCE CS2_33 points fitted on one discharge predict three later discharges. In synthetic tests, 1C-calibrated points predict noiseless 0.5C/1.5C responses without refitting (Figure 7). BKBAgent achieves 1.14 mV on the cell and 0.91/0.94 mV on Regular tasks; GP-BKB gives 2.50 and 1.22/1.44 mV. These tests assess single-cell temporal prediction and within-range protocol transfer. Appendix K gives the protocols and Extreme results.

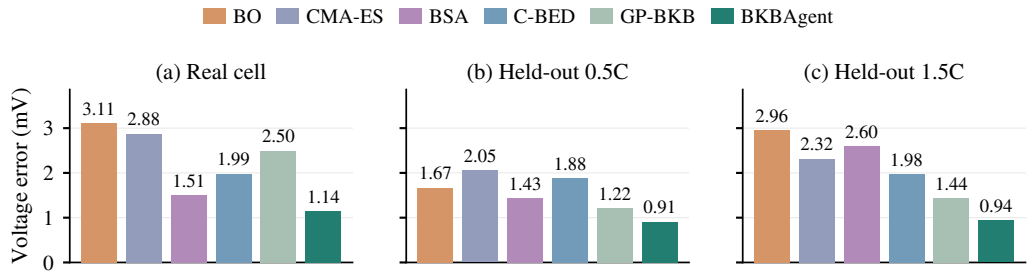

Figure 7: Frozen-point prediction. (a) One cell, three seeds and three subsequent discharges; time-weighted voltage RMSE. (b,c) Regular tasks: 10 parents, two parameterizations and three seeds; phase-mean voltage RMSE. All evaluations completed; bars show means.

5 Conclusion

BKBAgent combines physics-informed retrieval with joint-belief-guided simulation selection. Experiments support improved parameter recovery, with frozen-point predictions evaluated on unseen protocols and one measured cell. Expected belief change can nevertheless worsen reference agreement. Future work targets calibrated predictions, adaptive hypothesis support, and broader battery validation.

REPRODUCIBILITY STATEMENT

Appendices A–D specify the reasoning interfaces, response validation, evaluation protocol, and parent-level bootstrap. Appendix J gives the parameter domains, perturbation-generation rules, numerical and baseline settings, implementation scope, and KG sources. Appendix L prints the complete prompt templates, textual reply schemas, and context construction. Appendix K specifies the held-out and measured-cell protocols. Anonymous materials: `https://anonymous.4open.science/r/BKBAgent-07EB`.

AI USE STATEMENT

BKBAgent incorporates LLM-based components as part of the proposed method, as described in the methodology and Appendix L. We additionally used generative AI tools for manuscript editing, analysis-code development, and figure preparation. The authors review the resulting text, code, and figures for technical correctness and consistency with the reported evidence. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Weilong Ai, Ludwig Kraft, Johannes Sturm, Andreas Jossen, and Billy Wu. Electrochemical thermal-mechanical modelling of stress inhomogeneity in lithium-ion pouch cells. *Journal of The Electrochemical Society*, 167(1):013512, 2020. doi: 10.1149/2.0122001JES. URL https://doi.org/10.1149/2.0122001JES.

Antti Aitio, Scott G. Marquis, Pedro Ascencio, and David Howey. Bayesian parameter estimation applied to the Li-ion battery single particle model with electrolyte dynamics. *IFAC-PapersOnLine*, 53(2):12497–12504, 2020. doi: 10.1016/j.ifacol.2020.12.1770. URL https://doi.org/10.1016/j.ifacol.2020.12.1770.

Adrien M. Bizeray, Jin-Ho Kim, Stephen R. Duncan, and David A. Howey. Identifiability and parameter estimation of the single particle lithium-ion battery model. *IEEE Transactions on Control Systems Technology*, 27(5):1862–1877, 2019. doi: 10.1109/TCST.2018.2838097. URL https://doi.org/10.1109/TCST.2018.2838097.

Center for Advanced Life Cycle Engineering. CALCE battery research data. University of Maryland, n.d. URL https://calce.umd.edu/data. CS2 cell data; accessed September 26, 2026.

Chang-Hui Chen, Ferran Brosa Planella, Kieran O'Regan, Dominika Gastol, W. Dhammika Widanage, and Emma 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. URL https://doi.org/10.1149/1945-7111/ab9050.

Jiawei Chen, Xiaofan Gui, Shikai Fang, Shengyu Tao, Shun Zheng, Weiqing Liu, and Jiang Bian. Battery-Sim-Agent: Leveraging LLM-Agent for inverse battery parameter estimation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, 2026. doi: 10.1145/3770855.3818856. URL https://arxiv.org/abs/2605.29560.

Deepro Choudhury, Sinead Williamson, Adam Goliński, Ning Miao, Freddie Bickford Smith, Michael Kirchhof, Yizhe Zhang, and Tom Rainforth. BED-LLM: Intelligent information gathering with LLMs and Bayesian experimental design. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2508.21184.

Marc Doyle, Thomas F. Fuller, and John 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. URL https://doi.org/10.1149/1.2221597.

Joel C. Forman, Scott J. Moura, Jeffrey L. Stein, and Hosam K. Fathy. Genetic identification and fisher identifiability analysis of the Doyle–Fuller–Newman model from experimental cycling of a

LiFePO4 cell. *Journal of Power Sources*, 210:263–275, 2012. doi: 10.1016/j.jpowsour.2012.03.009. URL https://doi.org/10.1016/j.jpowsour.2012.03.009.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

David Greenberg, Marcel Nonnenmacher, and Jakob Macke. Automatic posterior transformation for likelihood-free inference. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2404–2414. PMLR, 2019. URL https://proceedings.mlr.press/v97/greenberg19a.html.

Nikolaus Hansen, Youhei Akimoto, and Petr Baudis. CMA-ES/pycma on github, 2019. URL https://doi.org/10.5281/zenodo.2559635.

Wei He, Nicholas Williard, Michael Osterman, and Michael Pecht. Prognostics of lithium-ion batteries based on Dempster–Shafer theory and the Bayesian Monte Carlo method. *Journal of Power Sources*, 196(23):10314–10321, 2011. URL https://web.calce.umd.edu/articles/abstracts/2011/11_He_prognostics_of_lithium_ion.html.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 132876–132907. Curran Associates, Inc., 2024. doi: 10.52202/079017-4224. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/efaf1c9726648c8ba363a5c927440529-Paper-Conference.pdf.

Apivich Hemachandra, Gregory Kang Ruey Lau, See-Kiong Ng, and Bryan Kian Hsiang Low. PIED: Physics-informed experimental design for inverse problems. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f7bdb4331e6e8af897f0c8891585e4f0-Abstract-Conference.html.

Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. GRAG: Graph retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 4145–4157. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.232. URL https://aclanthology.org/2025.findings-naacl.232/.

Mufei Li, Siqi Miao, and Pan Li. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/11e1900e680f5fe1893a8e27362dbe2c-Abstract-Conference.html.

Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mihaela van der Schaar. Large language models to enhance Bayesian optimization. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/84b8d9fcb4e262fcd429544697e1e720-Abstract-Conference.html.

Jan-Matthis Lueckmann, Jan Boelts, David Greenberg, Pedro Goncalves, and Jakob Macke. Benchmarking simulation-based inference. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 343–351. PMLR, 2021. URL https://proceedings.mlr.press/v130/lueckmann21a.html.

Rafael Oliveira, Dino Sejdinovic, David Howard, and Edwin V. Bonilla. Bayesian adaptive calibration and optimal design. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1800. URL https://papers.neurips.cc/paper_files/paper/2024/hash/673b9f8cba4ee9f15b88656a69761631-Abstract-Conference.html.

Kieran O'Regan, Ferran Brosa Planella, W. Dhammika Widanage, and Emma Kendrick. Thermal-electrochemical parameters of a high energy lithium-ion cylindrical battery. *Electrochimica Acta*, 425:140700, 2022. doi: 10.1016/j.electacta.2022.140700.

Stefan T. Radev, Ulf K. Mertens, Andreas Voss, Lynton Ardizzone, and Ullrich Köthe. BayesFlow: Learning complex stochastic models with invertible neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(4):1452–1466, 2022. doi: 10.1109/TNNLS.2020.3042395. URL https://arxiv.org/abs/2003.06281.

Tom Rainforth, Adam Foster, Desi R. Ivanova, and Freddie Bickford Smith. Modern Bayesian experimental design. *Statistical Science*, 39(1):100–114, 2024. doi: 10.1214/23-STS915. URL https://doi.org/10.1214/23-STS915.

Andreas Raue, Clemens Kreutz, Thomas Maiwald, Julie Bachmann, Marcel Schilling, Ursula Klingmüller, and Jens Timmer. Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics*, 25(15):1923–1929, 2009. doi: 10.1093/bioinformatics/btp358.

Tim Reichelt, Tom Rainforth, and Duncan Watson-Parris. Calibration of climate model parameterizations using Bayesian experimental design. *Machine Learning: Earth*, 2(1):015003, 2026. doi: 10.1088/3049-4753/ae2edb. URL https://doi.org/10.1088/3049-4753/ae2edb.

Silas Ruhrberg Estévez, Nicolás Astorga, and Mihaela van der Schaar. Timely clinical diagnosis through active test selection. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2879. URL https://papers.nips.cc/paper_files/paper/2025/hash/7c42e8a9eb471d225a4abec5e79d0153-Abstract-Conference.html.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0377. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Parshin Shojaee, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan K. Reddy. LLM-SR: Scientific equation discovery via programming with large language models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2404.18400.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, volume 25, pp. 2951–2959, 2012. URL https://papers.nips.cc/paper/2012/hash/05311655a15b75fab86956663e1819cd-Abstract.html.

Valentin Sulzer, Scott G. Marquis, Robert Timms, Martin Robinson, and S. Jon Chapman. Python battery mathematical modelling (PyBaMM). *Journal of Open Research Software*, 9(1):14, 2021. doi: 10.5334/jors.309. URL https://doi.org/10.5334/jors.309.

Nick Williard, Wei He, Michael Osterman, and Michael Pecht. Comparative analysis of features for determining state of health in lithium-ion batteries. *International Journal of Prognostics and Health Management*, 4:1–7, 2013. URL https://web.calce.umd.edu/articles/abstracts/2013/13_Comparative_Analysis_of_Features_for_Determining_State_of_Health_in_Lithium-Ion_Batteries.html.

Yinjiao Xing, Eden W. M. Ma, Kwok-Leung Tsui, and Michael Pecht. An ensemble model for predicting the remaining useful performance of lithium-ion batteries. *Microelectronics Reliability*, 53(6):811–820, 2013. URL https://web.calce.umd.edu/articles/abstracts/2013/13_An_ensemble_model_for_predicting_the_remaining_useful_performance.html.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

Algorithm 1 Knowledge-guided proposals, BED selection, and actual belief updating

Require: Measured responses Y^{obs}, protocols $\mathcal{P}$; model, parameter bounds, budget B
Extract observable features; ask the LLM for a query and response-node identifiers
Retrieve applicable connected evidence once and cache $\mathcal{K}$
Ask the LLM for complete parameter hypotheses and normalized initial belief b_0
Execute any declared warm starts within B and revise the belief from actual feedback
while budget permits evaluation of a complete parameter vector **do**
    Jointly generate $\mathcal{C}_t$ and candidate-specific outcomes $(y_m^{\mathrm{pred}}, p_m)$
    Compare each outcome: $\Delta_m = y_m^{\mathrm{pred}} - y^{\mathrm{obs}}$
    Assess $\widetilde{b}_t^{(m)}$ within BED while keeping b_t, H_t unchanged
    Compute Equation 3; select $\theta_t^\star$ and save the planning record
    Simulate $\theta_t^\star$ under $\mathcal{P}$ to obtain Y_t^{sim} and execution status
    Append the actual record to H_{t+1} with $\theta_t = \theta_t^\star$
    For a successful new execution, update b_{t+1} from Δ_t^{sim} and actual context
    Otherwise retain $b_{t+1} = b_t$; exclude hypothetical outcomes from both updates and history
    Update $\widehat{\Theta}_{t+1}$ if every calibration check passes; advance t
end while
return At most K verified complete parameter vectors

A Agent Reasoning and Execution

The same LLM carries out five reasoning stages: query construction, belief initialization, joint candidate generation and outcome prediction, hypothetical belief assessment, and belief updating from simulator feedback. Each request provides the relevant task context and distinguishes hypotheses about the target battery from chosen simulator inputs. The LLM returns structured JSON with physical-evidence identifiers and concise explanations; numerical code validates the response and executes the selected simulation. Algorithm 1 summarizes the loop, and Appendix L gives the prompts.

Reasoning stages. Query construction uses measured features, protocol context, and the available KG response-node identifiers. It returns a physical question, relevant node identifiers, and a protocol event. Belief initialization adds the retrieved evidence and parameter bounds to this context. The LLM proposes 2–24 distinct complete vectors in normalized coordinates, together with weights, explanations, and evidence identifiers. Coordinates are mapped to physical values using the specified linear or logarithmic transforms. These vectors form the fixed belief support; subsequent simulation candidates can lie elsewhere in the parameter domain.

A joint proposal-and-prediction call returns 2–8 complete candidates, each with 2–6 possible outcomes. Each outcome gives a probability and finite numerical values for the required observable features. At most 16 features are selected for prediction, prioritizing the response nodes identified by the query and distributing the remaining slots across records and stages. The complete measured feature collection remains available as context. Incomplete predictions are rejected before candidate assessment or simulation.

Hypothetical belief assessment receives the current belief, candidate–outcome pairs, their differences from the measurements, and physical evidence. For each outcome, the LLM assigns temporary weights to the same hypotheses, starting from the current belief independently. These assessments are stored in the planning record and used to compute BED utilities. Belief updating uses the actual execution history instead, with the new result identified explicitly. Its context contains the current belief, measurements, and cached knowledge, but excludes hypothetical outcomes and temporary weights. Both stages require explanations grounded in the supplied evidence and interpret simulation results through their agreement with measurements.

Response features and validation. Measured and simulated features use the same definitions within each protocol stage. When stage durations differ, simulated values are sampled at the measured stage's fractional positions, while actual durations are retained separately. Early and late windows retain physical time units; unresolved windows are omitted. CC/rest features describe volt-

age, and CV features describe current. Capacity and cutoff time are represented separately to retain their physical relationship. Full simulated traces are stored alongside the compact features used for reasoning.

Output acceptance uses the same response scorer for every method. It checks phase shapes and event durations separately, so matching a resampled shape is insufficient when the cutoff time is wrong. Features guide reasoning, while these full-response checks determine acceptance. The accepted archive is separate from the weighted belief and may contain vectors outside its support.

Probability normalization and consistency. Finite, nonnegative probabilities with positive total mass are normalized. Hypothesis weights receive a 10^{-8} floor and are normalized again to keep KL divergence finite. Validation rejects malformed responses, out-of-bounds vectors, duplicate proposals, unknown evidence identifiers, and changes to the hypothesis support. These checks enforce the response format and numerical constraints; they do not verify every physical claim in an LLM explanation. No model-generated code is executed.

For a fixed candidate $\theta_t^{(j)}$, let $\widetilde{w}_t^{(m)}$ denote the weights of $\widetilde{b}_t^{(m)}$, and w_t the current belief weights. Candidate dependence is implicit, as in Section 3.3. We measure consistency by

$$d_t(\theta_t^{(j)}) = \left\| \sum_{m=1}^{M} p_m \, \widetilde{w}_t^{(m)} - w_t \right\|_1 . \tag{4}$$

In a coherent Bayesian model with an exhaustive representation of outcomes, averaging conditional beliefs over those outcomes recovers the current belief, giving zero discrepancy. A finite outcome approximation or inconsistent LLM assessments can make it nonzero. We record this quantity as a diagnostic without adding it to the selection objective; Appendix E reports results for the saved mechanism-diagnostic subset. Utilities are computed as finite probability-weighted sums; no Monte Carlo standard error is estimated from repeated LLM samples.

Execution, failures, and stopping. Every attempted record solve counts against the budget, including failures. Failed attempts remain in the execution history but leave the belief unchanged. The utility is conditional on successful simulation and does not model failure probability. Exact cache hits require neither a new solve nor another belief update. If the LLM fails to update the belief after a completed simulation, that result and any validated archive entry are retained; the belief remains unchanged and the run stops with the recorded error.

Selection maximizes estimated utility, breaking ties by candidate identifier. Accepted vectors are ranked by the common weighted response discrepancy and retained greedily with a minimum Euclidean separation of 0.1 in normalized coordinates, up to the evaluation cap of five. The run records the budget used and the stopping reason: budget exhaustion, a round limit, no valid candidates, malformed output, or a client failure. Prompt and response logs omit provider credentials. Scripted tests check the interfaces and separation of hypothetical and actual records; prediction quality and identification performance are evaluated empirically.

B Physics-Informed Knowledge Graph

Each knowledge record identifies an entity or relation, its source, applicable models and materials, operating conditions, and dimensional assumptions. Equation–parameter links indicate which quantities participate in a physical relation. They specify neither a universal direction of parameter influence nor probabilistic conditional independence. Applicability conditions allow tasks with different material parameterizations to share physical relations while retaining their distinct constitutive functions.

The initial query uses measured features and the given protocol, without target parameter values or a preliminary simulation. Retrieved relations are cached throughout the run. A single fixed graph is shared across test tasks; inferred parameters, execution histories, and held-out responses are stored separately and never added to it. Each identification task starts with its own empty execution history.

Feature extraction preserves declared stage boundaries and marks those inferred from sampled current. Following the stage-level feedback used by BSA (Chen et al., 2026), the features include cutoff

durations, passed charge, and CC/CV proportions. Finer features describe early and late CC voltage slopes, CV initial and tail currents, CV decay rates, and early and late rest-voltage changes. Each retains its record, stage, units, time window, and operating context. Unresolved windows are omitted, and interpolation is restricted to the observed interval. Prescribed CC current and CV voltage are controls; retrieval and response comparisons use the resulting voltage and current responses, respectively.

Response–process edges indicate possible physical explanations. Early relaxation can involve both electrode diffusion and electrolyte transport, while CV initial current also relates to charge transfer and charge conservation. Model conditions allow the same response to retrieve different relations for DFN and SPM. Each retrieved equation is accompanied by its connecting paths, source, and participating quantities, including quantities that are fixed in the current task but remain relevant to interpreting the response.

Model coverage. The knowledge schema separates governing balances, constitutive laws, boundary and initial conditions, measurement operators, and executable parameter combinations. Its DFN view includes both electrodes, the separator, solid and electrolyte charge conservation, particle and electrolyte transport, electrode equilibrium potentials, interfacial kinetics, and terminal controls (Doyle et al., 1993; Sulzer et al., 2021). This is a curated isothermal, single-phase model view rather than the complete simulator expression tree. Thermal, degradation, plating, mechanics, hysteresis, and composite-particle options require separate scoped extensions. A graph parameter denotes a model quantity or constitutive function; only the task-declared coordinates are free identification variables. The six factors in Table 2 have explicit parameter bindings. They provide interpretable coupling evidence to the agent alongside the remaining physical relations.

Graph composition. We use a single knowledge graph containing 126 nodes and 341 edges: 25 response entities, 11 processes, 34 equation or physical-relation records, 36 parameters or constitutive quantities, 19 states, and one shared condition node. Its responses comprise four general response nodes and 21 more specific observable features. The six executable parameter combinations are a subset of the physical relations, not the total graph connectivity. Tasks share this graph and retrieve applicable subgraphs according to their model, material parameterization, and operating conditions.

Applicability checks cover temperature, initial SOC, material parameterization, and protocol modes. All stated conditions must hold, whether expressed as equality, set membership, or inclusive numerical intervals. Every record must satisfy the conditions attached to an equation and its linked condition nodes. Missing fields and unsupported conditions make applicability unknown and exclude the relation; an absent condition adds no restriction. This rule limits retrieval to documented applicability without claiming coverage of every physical regime.

To evaluate a parameter combination numerically, its graph record must provide product exponents, provenance, units, positive scalar quantities, and bindings to task parameters. An exponent may itself be a parameter, as in the Bruggeman law. Relations with unsupported numerical bindings can still provide sourced explanatory context, but are not compiled into numerical factors.

Mechanism factors and constitutive context. Table 2 lists the factors compiled from the shared graph. The symbols s_D, s_e, s_κ, and s_j denote positive multipliers of the corresponding reference functions. The model retains these functions rather than replacing them with constants. Factor preservation is defined at common state and temperature; it does not equate complete responses or imply non-identifiability of every listed parameter group.

A task's context specifies the material pair, cell family, parameterization, simulator version, and model options. It distinguishes absolute scalar parameters from constitutive-function multipliers and records function-source hashes. Chen2020 and ORegan2022 use the same KG with their respective constitutive functions supplied through this context. Conflicting model or material settings are rejected before inference. Measurement and protocol hashes, parameter ranges, and compiled relation identifiers are recorded before simulation; target parameter values are available only to the evaluator.

The initial belief combines the common physical ranges with LLM-generated joint hypotheses, providing a finite representation of the parameter domain. Matched knowledge ablations keep measurements, protocols, bounds, and prompts fixed while changing the supplied evidence.

Table 2: Source-linked factors under the spherical-particle and Bruggeman closures (Doyle et al., 1993; Sulzer et al., 2021). The capacity scale spans the full stoichiometric range of an electrode; it is not the usable full-cell capacity. Transport factors apply to DFN and SPMe, while the remaining families also apply to SPM.

Family	Compiled factor	Physical interpretation
Diffusion	R_s^2/s_D	Divided by $D_{s,\mathrm{ref}}$: diffusion time
Transport	$s_e\varepsilon_e^b$	Multiplier of $D_{e,\mathrm{ref}}$
Transport	$s_\kappa\varepsilon_e^b$	Multiplier of κ_{ref}
Capacity	$L\varepsilon_s c_{\max}$	Multiplied by $FA/3600$: capacity scale
Reaction	ε_s/R_s	Multiplied by 3: specific surface area
Reaction	$s_j\varepsilon_s/R_s$	Multiplied by $3j_{0,\mathrm{ref}}$: reaction scale

C Evaluation Definitions

For d estimated parameters with physical bounds l_j, u_j, point identification error is

$$\mathrm{NRMSE}_\theta(\widehat{\theta}, \theta^\star) = \left[\frac{1}{d}\sum_{j=1}^{d}\left(\frac{\widehat{\theta}_j - \theta_j^\star}{u_j - l_j}\right)^2\right]^{1/2}. \tag{5}$$

The metric uses physical parameter ranges even when the search operates in log coordinates. We select the point estimate with the lowest common calibration score among complete successful evaluations, breaking ties by execution ID. Runs with no complete evaluation receive a parameter NRMSE of one and remain in all aggregate statistics; their response errors are recorded as unavailable.

Response evaluation uses voltage during CC and rest phases, current during CV phases, and event durations. Let e_k denote the channel RMSE or absolute duration error for criterion k, with prespecified tolerance τ_k. The record score is

$$d(Y^{\mathrm{sim}}, Y^{\mathrm{obs}}) = \sqrt{\frac{1}{M}\sum_{k=1}^{M}(e_k/\tau_k)^2}. \tag{6}$$

Acceptance requires $e_k \le \tau_k$ for every criterion. The voltage, current, and duration tolerances are 10 mV, $0.01Q_{\mathrm{nom}}$ A, and $\max(2\,\mathrm{s}, 0.02T_k^{\mathrm{obs}})$, respectively, where the numerical value of Q_{nom} is in Ah. Evaluating phase shapes together with event timing prevents a small aggregate error from masking a missed criterion. Incomplete phases and unreached cutoffs fail validation.

All methods form archives from complete vectors evaluated within the simulation budget. Accepted vectors are sorted by calibration score and retained greedily if their Euclidean distance from every previously retained vector is at least 0.1 in unit coordinates, up to $K = 5$. Empty archives have size and Hit@5 zero. A Hit@5 success requires one retained vector satisfying $\max_j |\theta_j - \theta_j^\star|/(u_j - l_j) \le 0.1$, so all coordinates must meet the tolerance within the same vector. Minimum set error is reported only as a secondary evaluation metric; it does not determine the point estimate.

Point estimates and archives are fixed using calibration responses before accessing ground-truth parameters for scoring. The response errors in Table 10 refer to the specified calibration records; held-out prediction is evaluated separately in Appendix K. Results are averaged over seeds within each task, then over rates and parameterizations within each parent. We bootstrap independent parents 2,000 times to obtain intervals describing variation across test parents. Regular and Extreme contain ten and eight test parents, respectively; the ablation subset contains eight. Primary-benchmark summaries across modes weight the 18 retained parents equally.

Paired effects. For each comparator, the paired effect is BKBAgent's parent-level NRMSE minus the comparator's on the same parent. Within each mode, we resample its paired parents with replacement (ten Regular or eight Extreme) using 2,000 replicates and random seed 92401. The 2.5th and 97.5th percentiles of the mean differences define the interval; parameterizations, rates, and search seeds remain grouped within their parent. Table 3 reports pointwise intervals for BSA, C-BED, and

Table 3: Paired parent effects in parameter NRMSE. A negative difference favors BKBAgent. Rows use ten Regular or eight Extreme parents, with three seeds and six paired tasks per parent.

Mode	Comparator	Mean difference	95% paired CI
Regular	BSA	-0.0333	[-0.0367, -0.0302]
Regular	C-BED	-0.0370	[-0.0404, -0.0338]
Regular	GP-BKB	-0.0069	[-0.0075, -0.0063]
Extreme	BSA	-0.0376	[-0.0433, -0.0313]
Extreme	C-BED	-0.0334	[-0.0389, -0.0278]
Extreme	GP-BKB	-0.0056	[-0.0097, -0.0010]

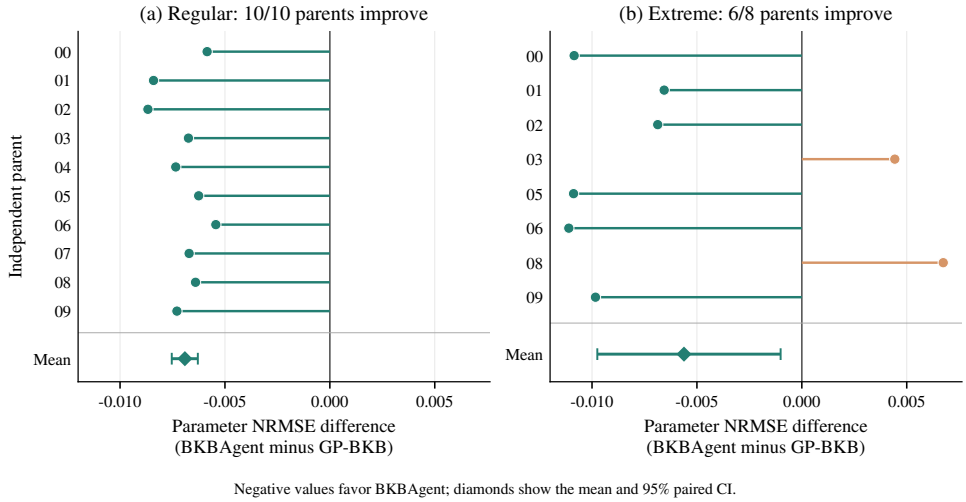

Figure 8: Paired parent differences against GP-BKB. Each point averages three search seeds, two parameterizations, and three calibration rates for one independent parent. Negative values favor BKBAgent. Diamonds show means over ten Regular or eight Extreme parents with 95% paired bootstrap intervals.

GP-BKB, without simultaneous coverage or p-values. The GP-BKB comparison extends the original two-comparator analysis. All six intervals lie below zero, supporting lower mean error over the evaluated parents while allowing individual tasks to favor a baseline. Figure 8 shows the individual parent differences against GP-BKB.

D CCCV Protocol and Paired Comparisons

Each cycle comprises a 1 s rest, CC charge to the upper cutoff, a 1 s rest, CV charge to C/20, a 90 s rest, CC discharge to 2.5 V, and a 90 s rest. The upper cutoff is 4.2 V for Chen2020 and 4.4 V for ORegan2022. The two cycles run continuously without resetting the state; SOC is set to 0.6 only before the first cycle. We score the second cycle using 57 channel-shape coordinates and three log-transformed event durations. Full traces retain physical time. Figure 4 shows the parent-00 cases selected before examining the final rankings.

The reporting exclusions in Appendix J.2 apply to all methods and paired protocols. All methods share target measurements, model context, parameter domains, calibration thresholds, and simulation budgets. Ground-truth parameters are accessible only to the evaluator after point estimates and archives have been fixed. One attempted two-cycle record counts as one solve, including failed records and final verification. The 16-task ablation subset uses the first four parents in each perturbation mode, both parameterizations, and 1C; Full reuses the corresponding primary runs. All methods repeat each task with seeds 92301, 92302, and 92303, chosen before inspecting results.

These seeds control local random choices such as initialization, while language-model sampling at the API provider may remain nondeterministic.

The frozen-point evaluation in Figure 7 uses 0.5C and 1.5C CCCV records with the same initial SOC, temperature, cutoffs, and continuous two-cycle execution. Parameters selected solely from each 1C calibration task are held fixed for both test rates. Unlike the calibration-response comparisons, these tests assess predictions on responses unused during identification. Appendix K specifies the noiseless references, scoring, and results for both perturbation modes.

E Prediction and Decision Diagnostics

Probabilistic outcome prediction. The history, hypothesis support, candidate input, and predicted outcomes are fixed before the actual simulator response is observed. Each observable feature has declared physical units and a scale determined on development data. Applying the same scaling to predicted features y_m^{pred} and actual features y_t^{sim}, we compute the energy score

$$\mathrm{ES} = \sum_{m=1}^{M} p_m \|y_m^{\mathrm{pred}} - y_t^{\mathrm{sim}}\|_2 - \frac{1}{2} \sum_{m,n=1}^{M} p_m p_n \|y_m^{\mathrm{pred}} - y_n^{\mathrm{pred}}\|_2. \tag{7}$$

Smaller values indicate better probabilistic predictions (Gneiting & Raftery, 2007). The score evaluates the finite set of possible outcomes and their probabilities directly, without requiring a continuous predictive density. Probability-weighted feature MAE provides a complementary measure of prediction error. Neither score alone establishes calibration.

The matched prediction comparison uses the parent and parameterization combinations from the 16-task ablation subset, evaluated at 0.2/1/2C with three seeds and two common candidate inputs per snapshot. Full, No external knowledge, and Text-RAG receive identical measurements, history, belief, and candidates. This comparison isolates evidence supplied during prediction while retaining knowledge already represented in the common belief. The end-to-end ablation evaluates the broader effect of external evidence. Text-RAG and KG retrieval share sources, model and material applicability filters, and context budgets.

Evaluation of simulation selection. Four mechanism tasks are evaluated with three seeds, checkpoints after 20 and 80 attempted solves, and four candidates per checkpoint. The with BED and w/o BED selectors commit their choices before the outcomes of unselected candidates are computed. Evaluating all branches requires at most 96 additional record solves, with exact cached responses reused. Each branch starts from a separate copy of the same belief and history and updates it using that candidate's actual response. Selectors that choose the same candidate share its branch update. We evaluate uniform selection by averaging all four branches. These diagnostic solves are counted separately, and their extra outcomes do not enter online agent memory.

For each free coordinate, the cumulative distribution function (CDF) gives the probability at or below a parameter value. Let $D(b, p_{\mathrm{ref}})$ be the maximum absolute difference between the belief and independent conditional-reference CDFs, taken over both free coordinates and all parameter values. We measure the effect of actual feedback by

$$\Delta D = D(b_t, p_{\mathrm{ref}}) - D(b_{t+1}^{\mathrm{actual}}, p_{\mathrm{ref}}). \tag{8}$$

Positive values indicate improved agreement with the reference; negative values indicate that feedback moves the belief farther from it. Unlike the nonnegative KL objective used for selection, this diagnostic measures improvement relative to an external reference and can have either sign for any selector or mechanism. It assesses belief quality separately from information gain and parameter recovery. A detrimental branch identifies an update that reduces reference agreement. Consistently negative means across all selectors would instead warrant examining shared prediction and updating components, rather than attributing progress to BED.

No external knowledge removes retrieved evidence while retaining the LLM's pretrained knowledge. The w/o BED variant changes only candidate selection: it retains KG evidence, belief, joint candidate generation and outcome prediction, simulator access, and actual-feedback updates throughout the iterative search. This differs from one-shot LLM inference. The BSA comparison evaluates agent backbones, BO, CMA-ES, and default parameters (Chen et al., 2026); our selection ablation

Table 4: End-to-end ablations on the matched 16-task subset. All configurations share response validation and the output cap. Brackets are parent-bootstrap intervals.

Configuration	Parameter NRMSE	Valid (%)
BKBAgent	0.090 [0.081, 0.100]	93.8
No external knowledge	0.109 [0.103, 0.114]	85.4
Text-RAG	0.114 [0.101, 0.130]	87.5
w/o BED	0.115 [0.100, 0.130]	83.3

Table 5: Utility decomposition, ranking sensitivity and actual joint-reference improvement on 24 matched snapshots. R/U is averaged over selected candidates. Agreement is with the original KL choice. Uniform averages all four branches and has no single top-1 choice. Positive ΔD_{joint} indicates improved reference agreement.

Selector	U	J	R	R/U	Top-1 match	ΔD_{joint}
Original KL	0.4696	0.3128	0.1568	0.3156	100.0%	0.0873
JSD	0.4621	0.3180	0.1442	0.2816	87.5%	0.0839
Drift only	0.4604	0.3009	0.1594	0.3287	87.5%	0.0908
LLM	0.2638	0.1880	0.0758	0.2384	25.0%	0.0733
Uniform average	0.2361	0.1749	0.0612	0.1918	–	0.0743

tests BED within the proposed agent. Uniform selection is confined to the matched branch diagnostic. GP kernels and numerical sampling moves belong to the numerical references and are outside the scope of these ablations.

Consistency of conditional beliefs. We recompute the discrepancy in Equation 4 from the saved outcome probabilities and conditional weights for all 96 candidates in the 24 mechanism-diagnostic snapshots described above. Each candidate has three predicted outcomes over a common support of 16 hypotheses. The mean discrepancy is 0.2161, the median is 0.1798, and the range is 0.0248–0.7332. These are descriptive candidate-level summaries of this diagnostic subset, not estimates for the full identification benchmark. The nonzero discrepancies show that the finite outcome assessments do not exactly recover the current belief. They can reflect outcome discretization or inconsistent conditional assessments, reinforcing the approximate interpretation of the KL utility.

Utility decomposition and selection sensitivity. For one candidate, write $b = b_t$, $q_m = \widetilde{b}_t^{(m)}$ and $\bar{q} = \sum_m p_m q_m$. On the common hypothesis support, Equation 3 decomposes exactly as

$$U = \underbrace{\sum_m p_m D_{\mathrm{KL}}(q_m \| \bar{q})}_{J\text{ : branch discrimination}} + \underbrace{D_{\mathrm{KL}}(\bar{q} \| b)}_{R\text{ : belief drift}}. \tag{9}$$

The first term is the probability-weighted Jensen–Shannon divergence. The second is nonzero when the outcome-weighted conditional beliefs do not recover the current belief. Identical q_m give $J = 0$ even if $R > 0$. Thus U measures expected belief change, not solely discrimination among possible outcomes. Replacing U by J removes this drift term but does not establish calibration or a coherent Bayesian model relative to b.

We recompute both terms for all 96 saved candidates and select the largest U, J, or R within each snapshot. LLM selection uses its recorded choice; uniform selection averages the four recorded branches. Each selected branch is evaluated using its actual-feedback weights and the joint-CDF distance defined in Appendix F. These additional selectors are offline sensitivity analyses on the same stored branches, not new end-to-end runs. The 24 snapshots comprise four conditional tasks, three seeds and two checkpoints; they are not 24 independent tasks.

For candidates selected by the original utility, mean R/U is 0.3156. Removing R changes the top-ranked candidate in 3/24 snapshots, all at the earlier transport checkpoint. The mean actual joint-reference improvement is 0.0873 for the original utility, 0.0839 for J, and 0.0908 for R alone. The drift-only result prevents attributing the observed improvement exclusively to branch discrimination; the comparison also does not support assuming that removing drift improves selection. These

Table 6: Joint dependence at 80 attempted solves. Entries are mean joint-CDF distances over three seeds for each two-parameter conditional task; lower is better. The independent product preserves both weighted marginals exactly. The last row weights the four tasks equally.

Family	Joint belief	Product of marginals	Product − joint
Diffusion	0.1663	0.3603	0.1940
Transport	0.2977	0.3476	0.0500
Capacity	0.2227	0.2924	0.0697
Reaction	0.2219	0.3460	0.1241
Mean	0.2271	0.3366	0.1094

descriptive one-step results do not determine how a changed policy would perform over a full identification run.

F Conditional Reference Distributions and Complete Vectors

Four reference tasks cover diffusion, transport, capacity, and reaction. Each estimates two parameters with the other eight fixed at known nominal values, defining a conditional problem rather than a marginal of the ten-dimensional task. Reference integration starts at 32^2 points, refines to 64^2, and permits 128^2 if needed. Convergence requires changes in marginal CDFs and interval endpoints below 0.02 in their respective probability and unit-coordinate scales, and an absolute log-evidence change below 0.05. A separate 72×72 grid is used to render the contours.

The registered reference target is $p_{\mathrm{ref}}(u \mid y, \theta^0_{-S}) \propto p_0(u) L(y \mid T(u), \theta^0_{-S})$, where u denotes the two free unit coordinates, T applies their linear or logarithmic parameter transforms, and θ^0_{-S} fixes the remaining coordinates. The prior is uniform in the unit coordinates, giving a log-uniform physical prior for logarithmically transformed parameters. The likelihood uses the DFN response under the 1C CCCV protocol and independent Gaussian errors in the registered phase samples and log event durations: 0.005 V for voltage, $0.005 Q_{\mathrm{nom}}$ A for CV current, and 0.01 for log duration. These observation-noise scales are distinct from the GP residual scales in Appendix J.3. Grid masses are normalized products of prior density, likelihood, and quadrature cell area in unit coordinates.

A diagnostic of joint dependence. Marginal CDF distances assess each parameter separately and can miss an incorrect dependence structure. We retain that diagnostic and additionally compute

$$D_{\mathrm{joint}}(b, p_{\mathrm{ref}}) = \sup_{x,y} \left| F_b(x,y) - F_{\mathrm{ref}}(x,y) \right|, \qquad F_b(x,y) = \sum_i w_i \mathbf{1}[x_i \le x,\ y_i \le y]. \tag{10}$$

For the discrete reference grid and weighted hypothesis support, this distance is evaluated on the union of their coordinate values, including support points between reference-grid coordinates. Actual branch improvement is $D_{\mathrm{joint}}(b_t, p_{\mathrm{ref}}) - D_{\mathrm{joint}}(b_{t+1}^{\mathrm{actual}}, p_{\mathrm{ref}})$.

To isolate dependence while preserving each weighted marginal, we compare $b = \sum_i w_i \delta_{(x_i, y_i)}$ with

$$b_\perp = b_x \otimes b_y = \sum_{i,j} w_i w_j \delta_{(x_i, y_j)}. \tag{11}$$

This product retains the same possible values and marginal probabilities for each coordinate but removes their association. Permuting coordinates of nonuniformly weighted particles would not generally preserve those marginals. The product can have up to 16^2 support points and is used only as a distributional diagnostic. Its cross-combinations are not validated simulator outputs or a matched-budget online agent, and we do not filter them in a way that changes their marginals.

The original joint belief is closer to the finite-resolution conditional reference in all four tasks, with mean distance 0.2271 versus 0.3366 for the product. This supports the descriptive value of the represented dependence in these cases, without establishing full-dimensional posterior calibration or a causal advantage in end-to-end search.

The complete-vector heatmaps order parameters as follows: positive particle radius; solid diffusivity multiplier; electrolyte diffusivity multiplier; electrolyte conductivity multiplier; positive elec-

Table 7: Conditional-reference diagnostics for the numerical comparison methods. Ref. grid status records convergence of reference integration. Reference CDF error measures each method's disagreement with the shared simulator-based reference.

Family	Ref. grid status	Reference CDF error	
		GP-BKB	C-BED
Diffusion	Converged	0.0062	0.0218
Transport	Converged	0.0360	0.0264
Capacity	Converged	0.0070	0.0249
Reaction	Converged	0.0113	0.0387

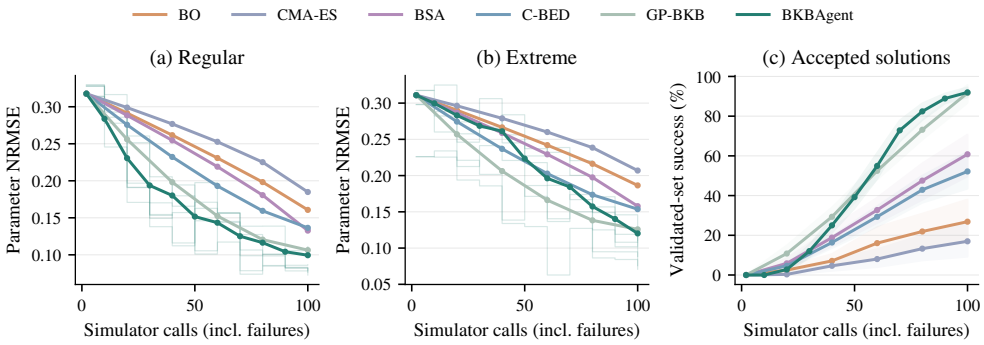

Figure 9: Progress against simulator calls, including failures. Regular and Extreme tasks are shown separately; faint staircases show six predetermined BKBAgent run trajectories per mode (parents 00–02, both parameterizations, 1C, the first repeated run). The final panel shows validated-set discovery with parent-bootstrap bands. The primary comparison remains the shared 100-call budget.

trode porosity; electrolyte Bruggeman exponent; positive active-material fraction; positive electrode thickness; maximum positive-electrode concentration; and exchange-current multiplier. Each row is a complete joint configuration. Entries show the bounded coordinate $v_j = (\theta_j - l_j)/(u_j - l_j)$ for linear parameters or $v_j = \log(\theta_j/l_j)/\log(u_j/l_j)$ for logarithmic parameters. Thus, equal coordinates can represent different physical values. Gray cells contain the eight fixed nominal coordinates; the numerical color scale applies only to the two estimated coordinates. Each belief marker represents a complete hypothesis projected onto these two coordinates, with area proportional to its normalized weight. The fixed support gives initial and updated markers identical positions. Archive membership depends on actual response validation independently of belief weight, and each reported solution retains one complete row.

The result summaries report convergence for all four reference integrations under the prespecified refinement tolerances. Both numerical methods are compared with the same finite-resolution reference; convergence status describes the reference integration, whereas CDF error compares their agreement with it. The new joint-CDF diagnostics use the normalized exported reference grid and do not independently establish convergence to a continuous reference. Agreement with this conditional reference does not establish a Bayesian posterior interpretation or ten-dimensional calibration for the LLM-elicited belief.

G Simulation Efficiency and Computational Cost

Figure 9 reports parameter recovery and validated-set discovery as functions of the simulation budget. One call is an attempt to simulate a complete two-cycle record, including initialization, failed integration, and uncached verification. The incumbent minimizes response error, so its parameter error may improve, plateau, or increase between checkpoints. Table 8 separates solver time, numerical search and selection overhead, and API workload. Requests and tokens include query construction, belief initialization, candidate generation and outcome prediction, BED assessment, and actual-feedback updating.

Table 8: Computation and API workload per run. Entries are means; solver time also reports the standard deviation across runs. Numerical time excludes API calls. Input and total tokens are in millions; output tokens are in thousands. Repeated context is counted for each request.

Method	Solver (s)	Numeric (s)	Requests	Input (M)	Output (k)	Total (M)
BO	633 ± 276	18.77	0.00	0.00	0.0	0.00
CMA-ES	560 ± 246	0.37	0.00	0.00	0.0	0.00
BSA	678 ± 299	0.08	80.00	2.12	50.4	2.17
C-BED	626 ± 272	48.92	0.00	0.00	0.0	0.00
GP-BKB	590 ± 259	53.73	98.00	1.57	76.9	1.65
BKBAgent	640 ± 280	0.22	295.59	2.38	189.7	2.57

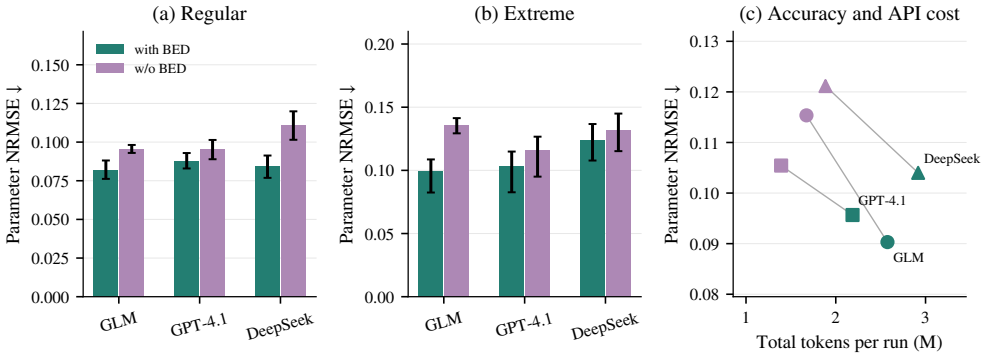

Figure 10: Backbone sensitivity on matched tasks. (a,b) with BED versus w/o BED in Regular and Extreme tasks, with parent-bootstrap intervals. (c) Identification error against total tokens per run; connected markers pair the two selection rules for one backbone. Marker shape identifies the backbone and color identifies the selection rule.

Solver time varies with protocol duration, parameter-dependent stiffness, and failed integration paths, even under equal call budgets. The updated usage records average 295.59 requests and 2.57 million tokens per BKBAgent run, compared with 80.00 requests and 2.17 million tokens for BSA, and 98.00 requests and 1.65 million tokens for GP-BKB. BKBAgent therefore uses 3.69× the requests and 1.18× the tokens of BSA, and 3.02× and 1.56× those of GP-BKB, respectively. Total tokens include input and output usage, including cached input regardless of billing.

H Sensitivity to the Language-Model Backbone

We assess backbone sensitivity by replacing the language model in every reasoning call. GLM-4.6V, GPT-4.1, and DeepSeek-V4-Pro are compared on the same 16-task subset with three paired search seeds. Pairing with BED and w/o BED within each backbone isolates the selection rule while holding the language model fixed. The GLM configurations reuse their matching ablation runs; the other two backbones contribute 192 runs. Their API identifiers are `glm-4.6v`, `gpt-4.1-2025-04-14`, and `deepseek-v4-pro`, respectively. DeepSeek's optional thinking mode is disabled.

All configurations share measurements, numeric observable features, KG contents and retrieval interfaces, prompts, context selection, parameter bounds, hypothesis and predicted-outcome counts, and simulation budgets. They use the same structured text interface, with no provider-specific image inputs or external tools. Temperature is 0.2 where supported, each request permits at most 8,192 output tokens, and at most two retries are allowed. The result export records model identifiers, providers, token counts, retries, and rejected outputs. Search seeds are paired, but provider-side sampling may remain nondeterministic; tokenization and latency can also differ across providers.

Figure 10 shows lower mean identification error with BED for all three backbones in both modes. Averaged across modes, the reductions relative to w/o BED are 21.7% for GLM-4.6V, 9.3% for

Table 9: Backbone comparison with API cost and format reliability. Workload for the reused GLM with-BED runs uses the same per-run records as Table 8. Invalid is the recorded schema-rejection count divided by the recorded request count.

Backbone	Selection	NRMSE	Tokens (k)	Invalid (%)
GLM-4.6V	with BED	0.090	2576	0.56
GLM-4.6V	w/o BED	0.115	1673	0.62
GPT-4.1	with BED	0.096	2187	0.39
GPT-4.1	w/o BED	0.105	1391	0.40
DeepSeek-V4-Pro	with BED	0.104	2918	1.29
DeepSeek-V4-Pro	w/o BED	0.121	1885	1.30

Table 10: Response errors in physical units, averaged over independent parents.

Method	Voltage (mV)	CV current (mA)	Event time (s)
BO	10.30	24.51	15.83
CMA-ES	10.76	28.20	16.07
BSA	8.69	24.79	13.01
C-BED	9.02	28.26	13.48
GP-BKB	6.91	22.95	10.52
BKBAgent	6.41	21.39	9.76

GPT-4.1, and 14.2% for DeepSeek-V4-Pro. Token usage increases by 53.9%, 57.2%, and 54.7%, respectively. GLM gains more on Extreme tasks, whereas DeepSeek gains more on Regular tasks. GPT uses the fewest tokens, while GLM achieves the lowest mean error. BED thus improves identification across the tested backbones, with gains and costs that depend on the model and task. These comparisons are limited to the evaluated backbones and battery settings.

I Channel Errors and Operating Strata

Table 10 reports voltage, current, and event-timing errors in physical units. Table 11 compares parameter recovery across perturbation modes, parameterizations, and rates using paired parent identities. Together, they distinguish response agreement from parameter accuracy and show how performance varies across operating conditions.

Table 11: Parameter NRMSE by perturbation mode, parameterization, and rate.

Mode	Parameters	Rate	BO	CMA-ES	BSA	C-BED	GP-BKB	BKB
Regular	Chen2020	0.2	0.168	0.193	0.134	0.147	0.103	0.094
Regular	Chen2020	1	0.145	0.163	0.115	0.125	0.089	0.080
Regular	Chen2020	2	0.157	0.182	0.126	0.141	0.098	0.093
Regular	ORegan2022	0.2	0.173	0.196	0.159	0.127	0.147	0.143
Regular	ORegan2022	1	0.154	0.178	0.127	0.133	0.095	0.087
Regular	ORegan2022	2	0.168	0.197	0.136	0.146	0.105	0.100
Extreme	Chen2020	0.2	0.195	0.215	0.172	0.154	0.130	0.122
Extreme	Chen2020	1	0.165	0.183	0.146	0.137	0.110	0.103
Extreme	Chen2020	2	0.184	0.209	0.164	0.151	0.120	0.117
Extreme	ORegan2022	0.2	0.207	0.228	0.184	0.179	0.136	0.129
Extreme	ORegan2022	1	0.181	0.198	0.158	0.141	0.119	0.112
Extreme	ORegan2022	2	0.187	0.209	0.122	0.158	0.140	0.138

J Reproducibility Settings

J.1 Parameter Domain and Numerical Solver

Table 12 lists the ten identification coordinates in their registered order. Both parameterizations describe the same LG M50 material system (Chen et al., 2020; O'Regan et al., 2022). Unlisted parameters and constitutive functions retain their values in the corresponding parameterization. The four multipliers scale the reference functions while preserving their state and temperature dependence. For bounds (ℓ_j, u_j), a unit coordinate $z_j \in [0, 1]$ maps to $\ell_j + z_j(u_j - \ell_j)$ for a linear coordinate and $\ell_j(u_j/\ell_j)^{z_j}$ for a logarithmic coordinate. Parameter NRMSE is computed in physical coordinates and normalized by the physical range $u_j - \ell_j$.

The numerical specification uses PyBaMM 25.6.0 and CasADi 3.6.7. The isothermal DFN model has spherical Fickian particles, one particle phase, and Bruggeman transport efficiency. The CasADi solver runs in `safe` mode with absolute and relative tolerances of 10^{-6} and a maximum integration interval of 60 s. Each of x_n, x_s, x_p, r_n, r_p has 20 mesh points on uniform one-dimensional submeshes with finite-volume discretization; the current collector is zero-dimensional. Each record begins at SOC 0.6 and 298.15 K and runs two continuous CCCV cycles. Scoring uses the second cycle, as detailed in Appendix D. Event-controlled steps have a maximum duration of 86,400 s. An incomplete or failed step invalidates the record; its trace is not padded.

J.2 Parent Generation and Perturbations

The registered task specification defines perturbations in the unit coordinates of Table 12. Regular parents draw all ten coordinates independently from $\mathrm{Uniform}(0.15, 0.85)$. Extreme parents change one coordinate to 0.05 or 0.95, selecting the endpoint farther from its nominal unit coordinate and breaking ties using the parent random stream; the remaining coordinates stay nominal. The original test parent indices $0, \ldots, 9$ select the corresponding perturbed coordinate. The four development parents use coordinates 1, 5, 8, and 10 in the table's one-based order.

Each parent shares the same unit-coordinate truth across Chen2020 and ORegan2022 and the three calibration rates, giving six paired tasks. The corresponding physical values use each parameterization's bounds. Generation assigns four development and ten test parent IDs per mode; all related tasks share their parent ID. The generation stream is PCG64, seeded by the first 128 bits of the SHA256 digest of the release, split, mode, parent index, attempt index, and purpose, concatenated in that order with a vertical-bar separator. Generation seeds and parameter truths remain evaluator-only and are separate from the three search seeds.

Reported test subset. We retrospectively exclude `test-extreme-04` and `test-extreme-07`, corresponding to coordinates 5 and 8, from all reported test aggregates. The construction forces these coordinates to the same endpoints in development and test

Table 12: Identification bounds, shared across methods. Only coordinate 9 has different native bounds between parameterizations. Lengths are displayed in μm for readability and stored in metres in the machine-readable configuration. The positive-electrode volume fractions must satisfy $\varepsilon_e + \varepsilon_s \leq 1$.

No.	Parameter (unit)	Lower	Upper	Nominal	Map
1	Positive particle radius (μm)	2.61	10.44	5.22	log
2	Positive diffusivity multiplier s_D	0.5	2	1	log
3	Electrolyte diffusivity multiplier s_e	0.5	2	1	log
4	Electrolyte conductivity multiplier s_κ	0.5	2	1	log
5	Positive electrode porosity ε_e	0.21775	0.335	0.335	linear
6	Electrolyte Bruggeman exponent b	1.3	2.3	1.5	linear
7	Positive active-material fraction ε_s	0.49875	0.665	0.665	linear
8	Positive electrode thickness (μm)	56.7	113.4	75.6	log
9	$c_{\max}$, Chen2020 (mol/m^3)	47,328	78,880	63,104	linear
	$c_{\max}$, ORegan2022 (mol/m^3)	38,823.75	64,706.25	51,765	linear
10	Positive exchange-current multiplier s_j	0.5	2	1	log

Table 13: Method-specific settings under the common solve budget. Further numerical options are specified below; KG sources are given in Appendix J.5.

Method	Registered configuration
BO	Ax 1.1.0, SingleTaskGP with a Matérn kernel, qLogNEI with $q = 1$. The two common starts plus Sobol points give 20 initial evaluations counted against the budget. Acquisition optimization uses 256 Monte Carlo draws, 512 raw samples, and 20 restarts.
CMA-ES	`cma` 4.4.0, initial standard deviation 0.2, population 10, bounds $[0, 1]$, and no restarts. Initialize at the better common start. At budget exhaustion, evaluate only the remaining prefix without assigning missing fitness values.
BSA	GLM-4.6V and the native knowledge, memory, and multimodal correction workflow (Chen et al., 2026); two common starts plus 18 warm-up evaluations leave 80 search evaluations. Shared parameter bindings, CCCV execution, and phase/time scoring adapt the upstream workflow to this benchmark.
C-BED	Continuous-response GP, expected information gain, and SMC; raw-coordinate kernel only ($\rho = 0$), four Sobol candidates, and two common starts. It uses neither an LLM nor KG retrieval. This controlled baseline is not a reproduction of BACON.
GP-BKB	Earlier GP-based knowledge-guided reference: cached KG, joint LLM proposals, a raw/relation kernel mixture, numerical expected information gain, and SMC. It is distinct from the finite LLM-belief method studied here.
BKBAgent	GLM-4.6V, 16 joint hypotheses, four candidates, three outcomes per candidate, and at most 16 predicted observable features in the evaluation specification. One cached KG retrieval; LLM conditional assessments; numerical weighted-KL selection.

(0.05 and 0.95, respectively), with the other coordinates nominal, yielding repeated complete truths under the stated rule. This reporting filter applies uniformly to all methods, parameterizations, rates, and seeds, including the frozen-point extension; stored runs are unchanged. The retained test set has ten Regular parents (60 tasks) and eight Extreme parents (48 tasks); development retains 48 tasks and inference remains ten-dimensional. Coordinates 1 and 10 have tied endpoint distances; their realized development/test endpoint choices have not been verified, so we do not claim complete truth-vector separation for the remaining splits.

The registered feasibility filter accepts a parent only if its paired calibration and reserved-reference simulations complete with finite responses. A failure triggers a redraw of the entire paired family, for at most 20 attempts; exhausting this limit stops materialization rather than replacing individual methods or protocols. No capacity-change or response-size threshold is applied. The CCCV stage sequence, C/20 termination, and continuous two-cycle execution follow BSA (Chen et al., 2026); the parameter domains, parent construction, known initial SOC of 0.6, and task counts specify our benchmark adaptation. Reserved references belong to the extension defined in Appendix D; generating a reference is separate from evaluating identified parameters under that protocol.

J.3 Search, Baselines, and Language-Model Settings

All main searches use a budget of 100 attempted record solves, including failures and warm starts, with seeds 92301, 92302, and 92303. They share two initial vectors: the nominal vector and a scrambled Sobol vector. Verified output sets contain at most five vectors, with a minimum pairwise Euclidean distance of 0.1 in unit coordinates. Table 13 gives the method-specific configurations under the registered CCCV protocol.

The GP-based references use 256 particles and 128 outcome draws. Both raw and relation length scales are 0.4, the relation weight is 0.5 when enabled, and the solver nugget is 10^{-9}. The registered residual scales are 0.15 V for voltage, 5 A for current, and 1 for log duration. SMC uses an ESS threshold of 0.5 and four Metropolis–Hastings sweeps. Its global proposal probability is 0.2, and its executed-point proposal probability is 0.3 with standard deviation 0.03. Local proposals use standard deviation 0.12 with scale multipliers $1, 0.1, 0.01$. For BO and CMA-ES, failed solves receive objective 10^6, and finite objectives are capped at $10^6 - 1$. Reporting retains the uncapped scores.

The declared interface for the current LLM-BED method and backbone comparison uses structured text, temperature 0.2 where supported, an output limit of 8192 tokens, and a transport retry limit of

two. The model identifiers are `glm-4.6v`, `gpt-4.1-2025-04-14`, and `deepseek-v4-pro`; DeepSeek thinking mode is disabled. The inherited BSA/GP-agent specification uses temperature 0, `top_p`=1, and the same output limit. Backbone results include run-level retry and schema-failure counts. API workload is reported as requests and input/output tokens; the comparisons do not claim a wall-clock speed advantage.

J.4 Response Validation and Implementation Scope

Appendix L provides the complete system prompt, five stage instructions, textual reply schemas, and the construction of their input context. The controller supports 2–24 hypotheses, 2–8 candidates, and 2–6 outcomes per candidate. The evaluation specification fixes these counts at 16, 4, and 3, respectively. The generic command-line interface defaults to at most eight output vectors and 16,000 output tokens; reproducing the declared settings requires overriding these limits to five and 8192.

Runtime checks require complete vectors within bounds, unique IDs, finite response values, all required descriptors, valid evidence IDs, and fixed hypothesis support. Each candidate–outcome branch is assessed exactly once. Nonnegative probabilities are normalized, while hypothesis weights are assigned a minimum value of 10^{-8} and normalized again. Numerical code computes utilities from these validated assessments; model-generated utility scores do not determine selection. Reproduction separates simulator execution, scoring of saved outputs, and deterministic regeneration of tables and diagnostic figures. Language-model responses may vary across provider invocations even with fixed local search seeds.

J.5 Knowledge Graph Sources

The KG draws on the DFN governing model (Doyle et al., 1993), PyBaMM (Sulzer et al., 2021), the PyBaMM 25.6.0 DFN documentation and source files for lithium-ion geometry and Bruggeman transport, and stage-feedback definitions adapted from BSA (Chen et al., 2026). It contains 126 nodes and 341 edges, with no fitted target cases or test histories. Retrieval filters relations by model, material, and operating conditions before assembling a cached evidence packet, limited to three hops and 20,000 characters.

K Frozen-Point Prediction and Real-Cell Evaluation

Scope and common controls. We report point predictions for BO, CMA-ES, BSA, C-BED, GP-BKB, and BKBAgent under the frozen protocol `figure7-generalization-20260926-v1`. Each calibration uses seeds 92301–92303 and a budget of 100 attempted record solves, including failures. The selected point minimizes the applicable calibration score, with execution-ID tie breaking. Parameters, model settings, preprocessing, and selection rules are fixed before test evaluation; test responses do not enter prompts, retrieval, belief updates, or parameter selection. All results in this extension evaluate the selected point predictions.

Synthetic rate transfer. We reuse the 1C calibration runs for 18 retained test parents, ten Regular and eight Extreme, paired with Chen2020 and ORegan2022, applying the exclusions in Appendix J.2. The resulting 36 tasks yield 648 method–seed runs. Each selected point predicts separate 0.5C and 1.5C records, giving 1,296 point evaluations against 72 shared noiseless references generated from the corresponding original target parameters. Each record starts at SOC 0.6 and 298.15 K, executes two continuous CCCV cycles, and scores the second cycle with the parameterization-specific cutoffs in Appendix D. No new synthetic calibration is performed. These rates differ from the calibration rate of each reused run but lie within the benchmark's 0.2C–2C range; this is within-range protocol transfer on the same LG M50 cell family.

Matched phases are sampled at fractional positions $0, 1/8, \ldots, 7/8$, with the final rest endpoint also included. Voltage error is the arithmetic mean of the six CC/rest phase RMSEs, reported in mV; current error is CV-phase RMSE in mA. Event error averages absolute duration errors for CC charge, CV charge, and CC discharge. The joint score is the root mean square of the seven phase errors and three duration errors after tolerance normalization (10 mV, 50 mA, and $\max(2\,\mathrm{s}, 0.02T_k^{\mathrm{ref}})$). Shape alignment does not replace duration scoring. A joint score below one alone does not imply that every criterion passes its tolerance.

Table 14: Fixed CALCE CS2_33 records. Each test discharge is independently initialized.

Role	Source workbook
Calibration	`CS2_33_8_17_10.xlsx`
Test 1	`CS2_33_8_18_10.xlsx`
Test 2	`CS2_33_8_19_10.xlsx`
Test 3	`CS2_33_8_30_10.xlsx`

Measured-cell split and model. The real-data study uses one CALCE CS2_33 cell, a nominal 1.1 Ah cell with a $LiCoO_2$ positive electrode (Center for Advanced Life Cycle Engineering, n.d.; He et al., 2011; Xing et al., 2013; Williard et al., 2013). Table 14 gives the fixed records; all use sheet `Channel_1-006`, native `Cycle_Index=1`, and discharge `Step_Index=7`. Eligible records are ordered by recorded discharge time, with the first used for calibration and the next three for testing. Thus, filenames are record identifiers rather than assumed measurement dates. Eligibility fixes the charging-completion condition (terminal CV current at most 0.05 A), finite increasing sample times, gaps at most 60 s, and discharge-current magnitude 0.545–0.555 A; records are not selected by fitted voltage error.

All methods fit this cell from scratch using a shared isothermal DFN proxy derived from PyBaMM's Ai2020 graphite/$LiCoO_2$ parameterization (Ai et al., 2020; Sulzer et al., 2021). Nominal capacity is changed from 2.28 to 1.1 Ah, and electrode width is scaled by $1.1/2.28$ at fixed height and electrode count; this is an effective-area adaptation, not a measured CS2 geometry. Voltage limits are 2.7–4.2 V, while the original Ai2020 OCV endpoints are preserved. Each discharge independently starts at SOC 1 with uniform concentrations and assumed temperature 298.15 K. No aging state, voltage offset, or nuisance initial state is fitted. The same ten parameter roles are used with Ai2020-specific bounds: radius $[1.5, 6]\,\mu$m, porosity $[0.208, 0.32]$, Bruggeman exponent $[1.3, 2.3]$, active fraction $[0.465, 0.62]$, thickness $[51, 102]\,\mu$m, and maximum concentration $[37457.25, 62428.75]$ mol/m^3; the four constitutive multipliers lie in $[0.5, 2]$.

Real-data prediction and scoring. The 18 calibration runs produce one point per method and seed for three independent test discharges, totaling 54 point predictions. Observed current is replayed with positive discharge sign, linear interpolation between samples, and the first current held backward to the physical discharge start. Voltage is scored only at the original observed times, without smoothing, offset fitting, or an invented time-zero observation. For squared voltage residuals e_i^2 at times t_i, the per-record error is

$$E_V = 1000\sqrt{\frac{\sum_{i=1}^{n-1}(t_{i+1}-t_i)(e_i^2+e_{i+1}^2)/2}{t_n - t_1}} \quad \text{mV}. \tag{12}$$

The real-cell score first averages the three test-record errors, then the three search seeds. Current is an imposed input, so current prediction, event timing, and the synthetic joint score are inapplicable. Independent physical parameter ground truth is unavailable, so no real-cell parameter NRMSE is reported.

Aggregation, failures, and interpretation. Synthetic summaries average seeds within each task, then the two parameterizations within each parent, then parents equally within each mode and test rate. Regular is shown in the main figure and Extreme in the appendix. Figure 7 reports means without confidence intervals; the single real cell does not support an independent-cell population interval. Failed or incomplete predictions retain their planned denominators and have unavailable metrics; real-cell block scores require all three test records to complete. No failed tail is trimmed, no reference failure triggers parent replacement, and no test score is imputed as zero. Additional prediction solves are separate from the calibration budget. Both studies use PyBaMM 25.6.0 with CasADi 3.6.7, safe-mode tolerances 10^{-6}, `dt_max`=60 s, a 600 s record timeout, and no automatic retries. The synthetic study concerns two parameterizations of one cell family. The measured study concerns short-horizon prediction for one cell under a fixed, imperfect literature proxy; it is neither recovery of measured CS2 material parameters nor transfer of LG M50-fitted parameters across chemistries.

Table 15: Frozen-point voltage error (mV). Real-cell means average three test discharges within each of three seeds (3/3 complete blocks, 9/9 cycles per method). Each test rate has 60/60 complete Regular and 48/48 complete Extreme predictions per method: ten or eight parents, two parameterizations and three seeds, aggregated within parent before averaging parents. Real-cell errors are time-weighted; synthetic errors are phase means.

Method	Real cell	Regular 0.5C	Regular 1.5C	Extreme 0.5C	Extreme 1.5C
BO	3.115	1.667	2.960	2.258	3.441
CMA-ES	2.879	2.054	2.316	2.783	2.917
BSA	1.509	1.432	2.604	2.026	3.330
C-BED	1.986	1.878	1.977	2.557	2.548
GP-BKB	2.500	1.216	1.442	1.505	1.899
BKBAgent	1.139	0.911	0.938	1.104	1.401

Table 16: Held-out synthetic prediction by perturbation mode and test rate. V is phase-mean voltage RMSE; I is CV-current RMSE; Time is mean absolute event-duration error; Joint is the tolerance-normalized score. All summaries use the same paired parent aggregation as Table 15. Complete counts refer to method–task–seed predictions.

Mode	Rate	Method	V (mV)	I (mA)	Time (s)	Joint	Complete
Regular	0.5C	BO	1.667	33.157	19.633	0.347	60/60
Regular	0.5C	CMA-ES	2.054	40.733	22.855	0.426	60/60
Regular	0.5C	BSA	1.432	29.366	15.919	0.300	60/60
Regular	0.5C	C-BED	1.878	35.577	18.411	0.363	60/60
Regular	0.5C	GP-BKB	1.216	25.268	13.559	0.250	60/60
Regular	0.5C	BKBAgent	0.911	20.566	10.272	0.202	60/60
Regular	1.5C	BO	2.960	54.957	31.279	0.775	60/60
Regular	1.5C	CMA-ES	2.316	45.435	25.586	0.642	60/60
Regular	1.5C	BSA	2.604	48.941	28.164	0.700	60/60
Regular	1.5C	C-BED	1.977	38.526	21.657	0.537	60/60
Regular	1.5C	GP-BKB	1.442	32.498	16.244	0.415	60/60
Regular	1.5C	BKBAgent	0.938	29.755	14.641	0.364	60/60
Extreme	0.5C	BO	2.258	48.306	28.918	0.489	48/48
Extreme	0.5C	CMA-ES	2.783	56.903	32.996	0.581	48/48
Extreme	0.5C	BSA	2.026	43.203	26.138	0.462	48/48
Extreme	0.5C	C-BED	2.557	48.319	26.893	0.507	48/48
Extreme	0.5C	GP-BKB	1.505	34.260	18.373	0.339	48/48
Extreme	0.5C	BKBAgent	1.104	25.918	14.102	0.259	48/48
Extreme	1.5C	BO	3.441	71.318	41.192	1.037	48/48
Extreme	1.5C	CMA-ES	2.917	61.788	34.202	0.859	48/48
Extreme	1.5C	BSA	3.330	68.637	38.251	0.963	48/48
Extreme	1.5C	C-BED	2.548	55.202	31.688	0.791	48/48
Extreme	1.5C	GP-BKB	1.899	44.985	23.763	0.596	48/48
Extreme	1.5C	BKBAgent	1.401	34.592	17.157	0.454	48/48

Observed point-prediction results. Table 15 reports voltage errors for all five settings; Table 16 retains the other synthetic response metrics. All 1,296 reported synthetic predictions and 54 real-cell discharge predictions completed, so no mean excludes a failed result. BKBAgent has the lowest mean voltage error in each reported setting. Comparator ordering varies: BSA outperforms GP-BKB on this measured cell, whereas GP-BKB performs better on both synthetic test rates. These observations concern the fixed tasks and cell; they do not establish an ordering across untested cells or protocols.

L PROMPT DESIGN OF BKBAGENT

A complete LLM request combines a shared system message, a stage-specific instruction, the current task context, and a response format. The system message defines the agent's role and the distinction between hypothetical and executed outcomes. The instruction assigns the reasoning task, while the context supplies the measurements, physical knowledge, and current search state. The five stages use the same LLM and shared memory.

The system message, stage instructions, and response formats below are reproduced from the controller, with line wrapping adjusted for readability. Context blocks describe the fields filled at runtime; they are not additional instructions or fabricated execution examples. Appendix J.4 specifies the implementation and evaluation settings.

L.1 SHARED SYSTEM PROMPT AND MESSAGE ASSEMBLY

Every call contains a `system` message with the following fixed text and a `user` message with the stage-specific content.

Shared system prompt

```
You are a battery parameter identification agent.  Use the fixed measured
responses, their protocols, retrieved physical evidence and actual
execution history.  The target battery's unknown parameters differ from a
chosen simulator input.  Compare every hypothetical or actual simulation
outcome with the fixed measurements.  Hypothetical conditional weights
are read-only BED assessments, not belief updates.  Only an actual
execution can update the current belief.  A simulator input does not
establish target parameters.  Preserve coupled alternatives the
measurements cannot distinguish.  Respect supplied KG evidence IDs,
applicability, units and qualifications.  Probabilities are approximate
judgements, not exact Bayesian posteriors.  Never invent executed results.
Hypothetical outcomes are independent planning branches, never execution
history.  Return requested JSON only, with concise evidence-based
justifications.
```

The phrase "read-only" in this literal system message means that a hypothetical assessment must leave the current belief and execution history unchanged. We refer to this reasoning stage as *hypothetical belief assessment*.

The user message is a JSON object with four fields: `stage`, `instruction`, `context`, and `reply_schema`. Thus, the task instruction and reply format are only two parts of the request. The controller fills `context` with current values and serializes the object using `json.dumps`, with `ensure_ascii=False` and `allow_nan=False`. The response format consists of textual field descriptions, not a formal JSON Schema enforced by the API. The returned JSON object is checked by the controller.

Shared task context. After retrieval, the context contains the following information. The five prompt blocks show how each stage uses it and which fields are added.

Context block	Fields and contents
Task and parameters	`model`, `chemistry`, and `parameters`: model and material identifiers, ordered parameter definitions, bounds, and transformations. `coordinate_rule` specifies normalized coordinates and logarithmic interpolation.
Measured responses	`measured_observations`: observable features with values, units, records, stages, and time windows. `protocols`: record names, operating conditions, and protocol steps.
Physical knowledge	`knowledge`: retrieved subgraphs, source and applicability information, and compiled physical relations.
Current belief	`belief`: complete joint hypotheses, their weights and explanations, and the fixed-support policy.
Execution history	`executed_history`: actual attempts, parameters, statuses, and response comparisons. Raw response arrays are omitted from this prompt context.
Remaining budget	`remaining_record_solves`: the number of record solves still available.

The coordinate rule is `unit in [0,1], log=true interpolates logarithmically`. The belief records that its weights are approximate rather than an exact Bayesian posterior. Query construction precedes retrieval and therefore uses the smaller context specified below.

L.2 Retrieval Query

The first call translates observable features and protocol context into a physical question and relevant KG response-node identifiers. Retrieval then follows the corresponding physical relations.

Prompt for retrieval query

Stage: `query`

Input context (populated at runtime)

Battery model: `model` and `chemistry`.
Observations and protocol: `measurements` and `protocols`, including stage-specific features and operating conditions.
Available response nodes: `available_feature_ids`; the query must select from these identifiers.

Task instruction (`instruction`)

```
Construct a retrieval query from measured observations and protocol
context; do not diagnose parameters.
```

Response format (`reply_schema`)

```
{
  "text":  "Physical question",
  "feature_ids":  "Nonempty subset of available_feature_ids",
  "event":  "Protocol event or empty string"
}
```

L.3 JOINT BELIEF INITIALIZATION

After retrieval, the LLM uses the measured features and applicable physical relations to propose weighted joint hypotheses. Each `unit` vector follows the complete parameter order in Table 12.

Prompt for joint belief initialization

Stage: `initialize_belief`

Input context (populated at runtime)

Task grounding: Model and material identifiers, ordered parameter bounds and transforms, measured features, and protocols.
Physical evidence: `knowledge`, including retrieved relations, their sources, and applicability conditions.
Initial search state: The shared `belief` has no hypotheses and `executed_history` is empty; the remaining solve budget is provided.

Task instruction (`instruction`)

```
Form joint parameter hypotheses and relative plausibility, including
coupled alternatives.  Do not claim complete coverage.
```

Response format (`reply_schema`)

```
{
  "hypotheses":  "2--24 {id, unit:  complete normalized vector, weight,
explanation, evidence_ids}"
}
```

L.4 JOINT CANDIDATE GENERATION AND OUTCOME PREDICTION

This stage uses the current belief and observed simulation feedback to propose new parameter vectors together with their possible outcomes. Each prediction covers the selected observable features. The `outcomes` field below describes the list nested inside each candidate.

Prompt for joint candidate generation and outcome prediction

Stage: `propose_and_predict`

Input context (populated at runtime)

Fixed task information: Shared model, material, parameter, measurement, protocol, and physical-knowledge fields.
Current search state: `belief`, `executed_history`, and `remaining_record_solves`.
Prediction targets: `response_schema`: selected measured features with their identifiers, units, and windows; every predicted outcome must cover these identifiers.

Task instruction (`instruction`)

```
Jointly propose informative unexecuted configurations and predict their
possible joint descriptor outcomes.  Explain which coupled hypotheses
each candidate could distinguish and how its parameter changes motivate
the predictions.  Predict descriptors, not pointwise curves; keep
duration, capacity and phase behavior compatible.  Probabilities describe
hypothetical outcomes conditional on successful execution.  Return
neither utility scores nor updated beliefs.
```

Response format (`reply_schema`)

```
{
  "candidates":  "2--8 {id, unit:  complete normalized vector,
explanation, evidence_ids, outcomes}; [] if no valid proposals",
  "outcomes":  "2--6 {id, probability, responses:  [{id,value}] covering
every response_schema ID, explanation, evidence_ids} per candidate"
}
```

L.5 Hypothetical Belief Assessment

For each candidate and possible outcome, the LLM assesses how the outcome would change the plausibility of the current parameter hypotheses. Each assessment starts from the same belief. BED then computes Equation 3 from the returned weights and outcome probabilities.

Prompt for hypothetical belief assessment

Stage: `evaluate_bed`

Input context (populated at runtime)

Fixed task information: Shared model, material, parameter, measurement, protocol, and physical-knowledge fields.
Current search state: The same `belief` and actual `executed_history` for every hypothetical branch, with the remaining budget.
Candidates and outcomes: `candidate_outcome_pairs`: complete candidate vectors, predicted features, probabilities, explanations, evidence identifiers, and computed differences from the measurements.

Task instruction (`instruction`)

```
Assess conditional hypothesis weights solely for BED information-gain
scoring:  if this candidate produced this outcome, how would its
comparison with fixed measurements affect the plausibility of each target
hypothesis?  Use the SAME current belief independently for every branch.
Do not update current belief, accumulate branches, or treat predicted
outcomes as executed evidence.  Keep hypothesis IDs unchanged.
Arithmetic computes utility.
```

Response format (`reply_schema`)

```
{
  "assessments":  "One {candidate_id,outcome_id,weights:{every hypothesis
ID: probability},explanation,evidence_ids} per branch"
}
```

L.6 Belief Update from Simulator Feedback

After a successful simulation, the LLM revises the current weights using the executed response and its agreement with the measurements. This call is skipped for an incomplete or failed simulation. Predicted outcomes and hypothetical assessments are excluded from its context.

Prompt for belief update from simulator feedback

Stage: `revise_actual`

Input context (populated at runtime)

Fixed task information: Shared model, material, parameter, measurement, protocol, and physical-knowledge fields.
Current search state: The current `belief`, remaining budget, and `executed_history`, now including the completed simulation and its feature comparisons.
New evidence: `new_execution_id`, identifying the history record used for this update.

Task instruction (`instruction`)

```
Revise from the actual new execution and agreement with fixed
measurements.  Do not count history twice or use any hypothetical branch.
Produce an approximate reasoning belief, not an analytic posterior.
```

Response format (`reply_schema`)

```
{
  "weights":  "Object:  every hypothesis ID to probability",
  "explanation":  "Evidence-based revision",
  "evidence_ids":  "Supplied equation IDs"
}
```

Response validation. The controller validates vectors, identifiers, feature coverage, and probabilities before computing BED utilities. These templates retain the generic count ranges in the implementation; evaluation fixes 16 hypotheses, four candidates, and three outcomes. Appendix J.4 specifies the validation rules and evaluation overrides.

M LIMITATIONS AND FUTURE WORK

The controlled identification benchmark uses two parameterizations of the same LG M50 material system. Frozen-point tests at 0.5C and 1.5C assess protocol changes within the benchmark's rate range, rather than extrapolation beyond it. The measured-cell evaluation covers one CALCE CS2_33 cell and three subsequent discharges under a shared capacity-scaled Ai2020 proxy. It therefore supports a short-term case study, not population-level robustness to model mismatch or long-term degradation. Each method fits that cell separately; the experiment does not transfer M50-fitted parameters across chemistries, and measured-cell parameter ground truth is unavailable. The extension assesses point prediction, without establishing predictive coverage of the accepted parameter sets. Broader cell, chemistry, and electrochemical-model validation remains future work.

The joint belief uses a finite set of hypotheses whose locations remain fixed during a run. This allows candidate simulations to be compared over the same support, but weight updates alone cannot assign belief mass to a region omitted at initialization. LLM-generated outcome probabilities and conditional weights also need not form a coherent Bayesian model. As the transport diagnostic illustrates, a large estimated KL utility does not necessarily improve agreement with the simulator-based reference. Adaptive hypothesis support, calibrated outcome predictions, and more consistent conditional assessments are directions for further study. Their value should be assessed against independent references; lower belief entropy alone would not establish an improvement.