# LEARNING FROM FAILURE: A PHYSICS-DRIVEN SELF-IMPROVING AGENTIC FRAMEWORK FOR LIQUID ELECTROLYTE DESIGN

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Liquid electrolytes are foundational to advanced batteries, and their formulation critically affects battery performance, safety, and lifetime. However, designing viable electrolytes remains a longstanding industrial challenge, as it requires navigating an interdependent space of discrete components and continuous concentrations under complex, often conflicting electrochemical constraints. In this work, we introduce SAGE, a **S**elf-improving **AG**entic framework for **E**lectrolyte discovery that iteratively refines its design policies through physics-driven feedback without human supervision. Central to SAGE is a principled paradigm of self-evolving autonomy, in which the agent iteratively improves through molecular dynamics evaluations and distills case-specific failure experience into actionable guidance for subsequent formulation design. Moreover, SAGE establishes an auditable discovery trajectory that documents constraint violations, retrieved evidence, hypothesis revisions, and resulting policy updates as structured knowledge, providing a reusable blueprint for tackling new scientific problems. Experiments across diverse real-world electrolyte discovery scenarios show the effectiveness of SAGE and the importance of physics-guided self-improvement for designing feasible, novel formulations.

## 1 INTRODUCTION

Liquid electrolytes are essential to industrial batteries that power applications ranging from electric vehicles and portable electronics to energy storage systems (Qin et al., 2021; Giffin, 2022). Central to their function is electrolyte *formulation*, which governs how molecular-scale interactions translate into cell-level performance (Wang et al., 2018; Suo et al., 2015). The interplay among salts, solvents, and additives collectively shapes ion transport, solvation structure, and interphase formation, thereby influencing electrochemical stability and cycle life (Wan et al., 2024; Xiao et al., 2024). Such interdependence grows increasingly consequential as modern batteries are pushed toward fast charging, high voltage, and low temperatures, where formulations optimized for one regime often become bottlenecks in another (Song et al., 2025). Therefore, systematic electrolyte design is indispensable for next-generation battery technologies (Qin et al., 2021; Wang et al., 2023).

However, electrolyte design is nontrivial because formulation behavior is determined by highly coupled physicochemical factors (Xia et al., 2019; Ren et al., 2019a; Grey & Hall, 2020). This challenge is further complicated by a hybrid search space in which the choices of salts, solvents, and additives are *discrete*, whereas concentrations and mixing ratios are *continuous* and often subject to strict compatibility constraints (Yang et al., 2020; Zhai et al., 2020). Even minor compositional changes can produce pronounced shifts in electrochemical performance and cycle life (Ren et al., 2019b; Park et al., 2023; Sharma et al., 2023). As a result, the development of advanced batteries requires improved techniques for designing viable multicomponent formulations under competing constraints.

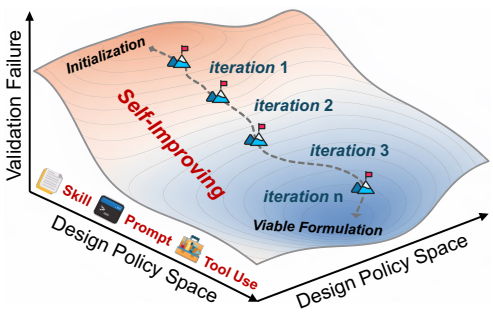



Figure 1: Overview of SAGE's design.



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.

Given these challenges, laboratory experiments have proven effective, but their trial-and-error nature makes them difficult to scale to high-throughput studies (Dave et al., 2022; Noh et al., 2024). More recently, data-driven methods have emerged to accelerate electrolyte design (Yan et al., 2025; Jain et al., 2026). Formulation graphs map electrolyte composition to device performance (Sharma et al., 2023), while chemical foundation models and unified predictive–generative models enable generative screening and constraint-aware mixture generation (Zohair et al., 2025; Yang et al., 2026). Yet these efforts remain constrained by limited data coverage, fixed protocols, and hand-crafted procedures, which bias them toward simpler, familiar recipes that fail to meet modern battery demands (Wang et al., 2025a). More fundamentally, the long-horizon search over the vast formulation space necessitates a self-directed scientific discovery loop involving hypothesis generation, candidate evaluation, and adaptive refinement of experimental strategies based on prior feedback. Consequently, this motivates a workflow that autonomously learns design principles from physics-informed evidence.

In this work, we propose SAGE, a self-improving agentic workflow for electrolyte formulation design. SAGE is built on the view that electrolyte discovery is not a one-shot generation task but a trial-and-error scientific search process, in which physical failure modes drive the autonomous evolution of design policies across iterations. This view is motivated by the complexity of searching for feasible electrolytes within a hybrid formulation space, in which scientific hypotheses, supporting evidence, and candidate formulations must be continually revised in response to feedback, and each round of physical evaluation can expose limitations in prior policies. However, manually coordinating this iterative discovery process is inefficient. We therefore seek to develop a training-free workflow capable of reflection and self-correction during task execution.

To this end, SAGE orchestrates agent reasoning and actions using failure feedback from each iteration to refine design policies for the next. In particular, it distills constraint violations from molecular dynamics simulations into actionable experience that informs structured updates to agent skills, task prompts, and tool-use strategies. To enforce scientific rigor, SAGE confines agents to reasoning over hypotheses, evidence, and design policies, while deterministic programs execute composition calculations and constraint checking, and molecular dynamics offers reliable physical feedback. This principled boundary allows agents to optimize scientific exploration while formulations are evaluated against objective criteria. Importantly, SAGE builds an auditable workflow by explicitly tracing how specific constraint violations drive revisions to hypotheses, evidence, and candidate formulations, while producing transferable artifacts for future tasks without model training or adaptation.

Empirically, we validate SAGE across diverse real-world battery development scenarios, with constraints grounded in experimental studies of transport, solvation, and interphase behavior (Mistry et al., 2022; Liu et al., 2025; Song et al., 2025). Ablation studies further show that physics-informed self-improvement is critical: disabling physics-driven feedback or replacing design policy updates with static text reflection substantially degrades performance, while the workflow remains competitive across base models of different sizes. Across these scenarios, SAGE identifies a diverse set of novel, compositionally complex formulations that combine multiple salts and functional additives to satisfy competing physicochemical constraints. Notably, SAGE outperforms state-of-the-art baselines, highlighting its practical potential for industrial electrolyte design. Moreover, structured artifacts accumulated by SAGE over multiple iterations effectively guide future runs of the same task, enabling markedly stronger performance as early as the second iteration. Our main contributions include:

- We frame liquid electrolyte discovery as a self-improving agentic search problem and propose SAGE, a training-free framework for navigating complex formulation spaces.
- We introduce a physics-driven failure learning workflow that distills feedback from molecular dynamics simulations into structured knowledge, enabling the iterative refinement of hypotheses, candidate formulations, and design policies without external supervision.
- We validate SAGE across diverse industrial electrolyte design scenarios, demonstrating that it substantially outperforms strong baselines and designs viable multicomponent electrolytes under complex physicochemical constraints.

## 2 METHOD

As shown in Figure 2, liquid electrolyte discovery is inherently a closed-loop design task that requires not only generating candidate formulations but also coordinating scientific tools and feedback-guided

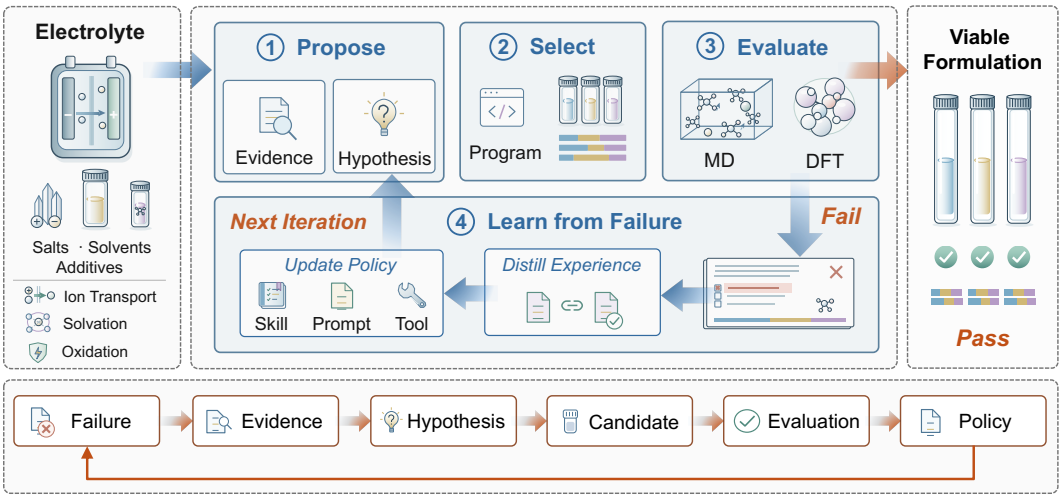



Figure 2: **Illustration of the SAGE workflow**. Liquid electrolyte design remains highly challenging due to its hybrid search space and competing constraints. To address this, we propose a physics-driven, self-improving agentic framework that distills failure experience to support iterative design policy refinement. Guided by physically grounded feedback, SAGE performs *self-driven correction* and continuously optimizes the design process through iterative adaptation. Building on this, SAGE enables autonomous improvement without external intervention or manual adjustment.

refinement across iterations. We thus propose an agentic workflow that integrates reasoning, tool use, and external interaction to update hypotheses and adapt design policies beyond one-shot generation. However, static agents remain insufficient, as effective exploration depends on systematic learning from failure signals rather than isolated reactions to individual trials, making a self-improving mechanism essential across iterations. Moreover, such refinement cannot rely on text-based reasoning or heuristics alone, since electrolyte performance is governed by highly coupled physicochemical properties that must be assessed objectively under the conflicting requirements of modern batteries. Building on this, SAGE grounds design policy refinement in physics-driven feedback, enabling rigorous and auditable exploration of formulation spaces within a training-free workflow.

### 2.1 Problem Setup

Electrolyte formulation combines discrete choices of salts, solvents, and additives with continuous concentrations and mixing ratios, creating a hybrid design space with both combinatorial and continuous complexity. These factors interact through coupled electrochemical properties, so improving one constraint may violate another. Consequently, SAGE begins with a frozen task specification

$$\mathcal{S}_t = (\Omega_t, \mathcal{O}_t, \mathcal{G}_t, \mathcal{B}_t)\,, \tag{1}$$

where $\Omega_t$ defines the pool of valid formulations, $\mathcal{O}_t$ the task-level objectives together with their optimization direction, $\mathcal{G}_t$ the set of feasibility constraints and physical-validation criteria that every candidate must satisfy, and $\mathcal{B}_t$ the available physical evaluation budget. Crucially, $\mathcal{G}_t$ includes only constraints that can be evaluated directly by molecular dynamics (MD) simulations or density functional theory (DFT) calculations. SAGE therefore does not interpret computational PASS results as evidence of long-term experimental performance.

For each valid discrete choice of component identities $z$, the feasible continuous variables, such as concentrations and mixing ratios, define a task-specific region $\mathcal{X}_t(z)$. Changing even a single component can thus alter which continuous settings remain valid for a given formulation. SAGE respects this coupling by letting agents propose scientifically meaningful discrete combinations, while deterministic programs handle the resulting composition search and constraint check within $\mathcal{X}_t(z)$.

At iteration $i$, the workflow generates a candidate set $\mathcal{C}_{t,i}$ and selects a smaller subset $\mathcal{K}_{t,i}$ for downstream physical evaluation. Each fully specified composition is assigned an order-invariant identity $h(c)$ so that the same formulation is not counted twice across iterations. Let $I_g(c) \in \{0, 1\}$ denote the binary verified outcome of gate (*i.e.,* a specific constraint) $g$ for candidate $c$. If $\mathcal{V}_{t,i-1}$ is

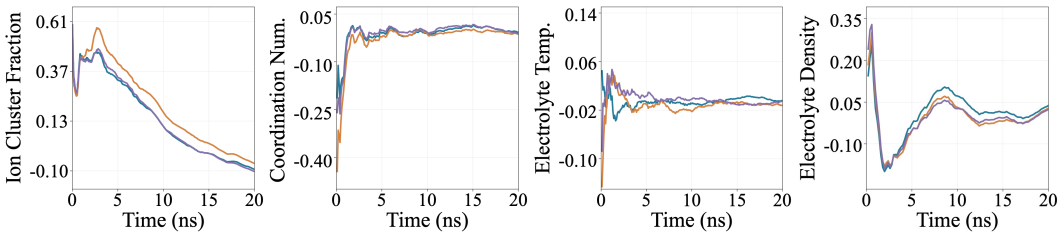



Figure 3: Convergence of three MD trajectories used to evaluate key physicochemical properties of the designed electrolyte formulations. Additional results are provided in Section A.7.

the set of unique valid identities found before iteration $i$, then the discovery objective is

$$\mathcal{V}_{t,i} = \mathcal{V}_{t,i-1} \cup \left\{ h(c) : c \in \mathcal{K}_{t,i},\ \prod_{g \in \mathcal{G}_t} I_g(c) = 1 \right\}. \tag{2}$$

That is, a newly proposed candidate is admitted into $\mathcal{V}_{t,i}$ in Equation (2) only if it passes every gate in $\mathcal{G}_t$ and has not been discovered in earlier iterations, thereby giving the search a precise objective under the finite budget $\mathcal{B}_t$. To let SAGE learn from failure signals, the state at iteration $i$ also carries a physical archive $\mathcal{A}_{t,<i}$ that records the constraint violations and evaluation outcomes accumulated so far, together with a versioned design policy $\pi_{t,i}$ that guides candidate proposal. Only $\pi_{t,i}$ can be updated across iterations, whereas $\mathcal{S}_t$, its constraints, and past outcomes remain immutable.

### 2.2 Evidence-Aware Agentic Search Workflow

Static agents can generate plausible formulations, but they lack a principled basis for deciding what to investigate after a constraint violation (Ruža & Gomez-Bombarelli, 2026). SAGE addresses this challenge by separating scientific exploration from numerical execution: agents retrieve evidence and formulate hypotheses, while deterministic programs generate compositions, enforce constraints, and allocate the simulation budget. This boundary ensures that each reasoning step produces actionable guidance for the next iteration of the search, rather than merely accumulating dialogue.

Concretely, SAGE orchestrates the workflow through five complementary agent roles. The *planner* initializes each iteration using the fixed objective and remaining budget, preventing intermediate results from changing the targets or resource allocation. The *retriever* gathers constraint-specific evidence by linking observed failures to relevant mechanisms, experimental conditions, and counterevidence. The *proposer* converts this evidence into testable hypotheses about component interactions and search directions, leaving concentrations and other numerical choices to deterministic programs. The *evaluator* performs deterministic constraint checking and bounded risk analysis before committing candidates to MD or DFT evaluation. The *reflector* converts evaluation outcomes into structured updates to the design policy $\pi_{t,i}$, including agent skills, task prompts, and tool-use strategies. These updates guide retrieval, hypothesis formation, and candidate search in the next iteration.

At the start of iteration $i$, $F_{t,i-1}$ denotes the set of task-specific failure logs carried over from iteration $i-1$, with $F_{t,0} = \emptyset$ at initialization. Each record contains the violated gate, the evaluated quantity and method, the chemical components, and the salt–solvent–additive context. Guided by $F_{t,i-1}$ and the task specification in Equation (1), the retriever collects evidence that can explain or help resolve the observed limitations, together with its source and conditions of applicability. If $F_{t,i-1}$ is empty, the gates in $\mathcal{G}_t$ define prospective queries rather than observed failures. The proposer then converts the retrieved evidence into qualitative, testable hypotheses. Each hypothesis specifies the target gate, relevant component families or interactions, the underlying electrochemical mechanism, and suitable search operators. It may incorporate counterevidence, but it never provides concentrations, predicted physical values, or pass/fail labels. The resulting information flow is

$$\begin{aligned} \mathcal{E}_{t,i} &= \mathrm{Retrieve}(\mathcal{S}_t, F_{t,i-1}, \pi_{t,i}), \\ \mathcal{Q}_{t,i} &= \mathrm{Propose}(\mathcal{S}_t, \mathcal{E}_{t,i}, \mathcal{A}_{t,<i}, \pi_{t,i}), \\ \mathcal{C}_{t,i} &= \mathrm{Compose}(\mathcal{S}_t, \mathcal{Q}_{t,i}, \mathcal{A}_{t,<i}, \pi_{t,i}, \xi). \end{aligned} \tag{3}$$

Here, $\mathcal{E}_{t,i}$ denotes the evidence bundle, $\mathcal{Q}_{t,i}$ denotes the set of qualitative hypotheses, and $\xi$ is a fixed random seed. Equation (3) captures the core logic of evidence-conditioned search: a physical failure changes what evidence is retrieved, the retrieved evidence updates the scientific hypothesis, and the hypothesis, in turn, guides which region of the formulation space should be explored further.

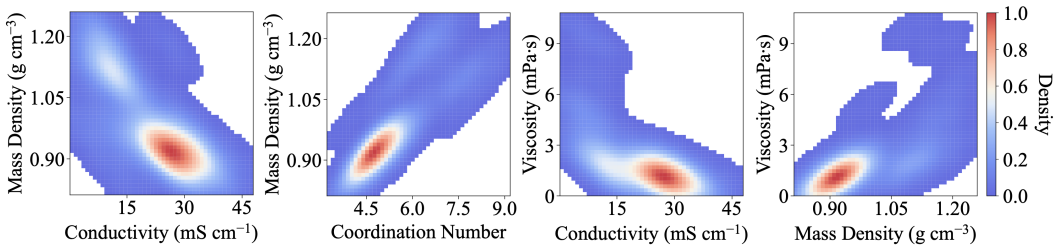



Figure 4: Key electrochemical properties of the designed formulations from MD simulations using the optimized potentials for liquid simulations (OPLS) force field (Jorgensen et al., 1996). Additional results are provided in Section A.7.

Given these hypotheses, the deterministic composer instantiates candidates within the feasible region $\mathcal{X}_t(z)$. The first iteration emphasizes broad coverage of $\mathcal{X}_t(z)$. Later iterations shift to local exploitation around valid formulations and treat failures near the boundary as viable opportunities, applying bounded concentration changes, role-preserving substitutions, and salt or solvent recombination. With $v_g(c) \geq 0$ denoting the normalized violation of constraint $g$ (zero when satisfied), the aggregate $V_t(c) = \sum_{g \in \mathcal{G}_t} v_g(c)$ distinguishes feasible anchors from informative near misses: a near miss is excluded from $\mathcal{V}_{t,i}$ but may still reveal a boundary worth exploring. The final subset $\mathcal{K}_{t,i}$ selected for physical evaluation balances this exploration signal with evidence support and compositional diversity, so that the resulting tests cover complementary hypotheses rather than near-duplicate candidates.

### 2.3 Physics-Driven Candidate Evaluation

Evidence and heuristic reasoning can generate plausible formulations, but they cannot establish whether the coupled physicochemical constraints of a candidate are satisfied under physical simulations. SAGE thus uses MD and DFT for physical evaluation. Candidates are frozen before evaluation, and the same gates are applied in every iteration, preventing changes to either a formulation or its pass criteria. For each formulation, SAGE builds a charge-neutral periodic system, equilibrates its density, and evaluates the resulting MD trajectory. For molecular species $s$, the mean-squared displacement (MSD) tracks how far its particles move on average over a lag time $\tau$:

$$\mathrm{MSD}_s(\tau) = \frac{1}{N_s} \sum_{j \in s} \left\langle \|\mathbf{r}_j(t_0 + \tau) - \mathbf{r}_j(t_0)\|^2 \right\rangle_{t_0},$$

where $N_s$ is the number of particles belonging to species $s$, $\mathbf{r}_j(t)$ is the unwrapped position of particle $j$ at time $t$, and $\langle \cdot \rangle_{t_0}$ denotes an average over all valid time origins $t_0$. From the MSD, we estimate the diffusion coefficient $\widetilde{D}_s$ and then compute a transport-based proxy for ionic conductivity:

$$\widetilde{D}_s = \mathrm{median}_{\tau \in [50,100]\ \mathrm{ps}} \frac{\mathrm{MSD}_s(\tau)}{6\tau},$$

$$\sigma_{\mathrm{NE}}^{\mathrm{screen}} = \frac{1}{V k_B T} \sum_s N_s q_s^2 \widetilde{D}_s,$$

where the sum runs over charged species with signed charge $q_s$, and $V$, $T$, and $k_B$ have their usual meanings. We emphasize that $\sigma_{\mathrm{NE}}^{\mathrm{screen}}$ is a screening proxy. Coordination and phase stability are read from physical feedback, while relative oxidation tendency is judged separately by DFT calculations on clusters extracted from MD. To determine whether a numerical gate is passed, SAGE uses a bootstrap. It resamples aligned trajectory blocks $B$ times, and for each resample $b$ computes the gate statistic $y_g^{*(b)}(c)$, where $g(y) = 1$ means the threshold is met. The estimated pass probability is

$$\widehat{p}_{\mathrm{pass},g}(c) = \frac{1}{B} \sum_{b=1}^{B} \mathbb{I}\left[ g\left( y_g^{*(b)}(c) \right) = 1 \right]. \tag{4}$$

A numerical gate is counted as passed ($I_g(c) = 1$ in Equation (2)) only if its quality-control criteria are satisfied, its estimated value meets the threshold, and $\widehat{p}_{\mathrm{pass},g}(c)$ exceeds a fixed confidence level. Boolean gates use direct verdicts rather than bootstrap probabilities. Evaluation may stop early for clear failures, while borderline candidates may be simulated longer within the fixed budget. In

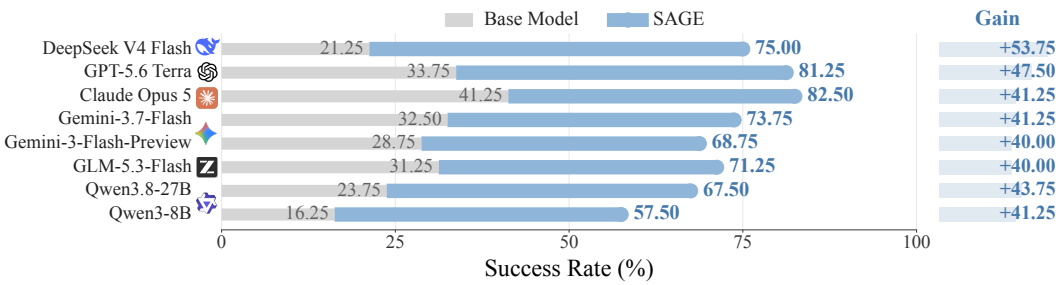



Figure 5: SAGE consistently improves formulation design success rates across diverse base models, with an average gain of 43.6 percentage points. Additional results are provided in Section A.7.

addition, a quality-controlled result is added to $\mathcal{A}_{t,\leq i}$, and a chemical threshold miss is recorded as a failure, whereas an execution failure is treated only as a process issue and is excluded from failure learning. Figure 3 illustrates SAGE's MD-based evaluation of convergence in key properties through three representative trajectories, while Figure 4 shows the corresponding property distributions.

### 2.4 Self-Improvement for Design Policy Refinement

Physical feedback is useful only if it shapes the next iteration in a controlled way. SAGE thus converts failure signals into structured updates to agent skill, task prompt, and tool-use strategy, forming the adaptive design policy $\pi_{t,i} = (s_{t,i}, p_{t,i}, u_{t,i})$. Since the base model and task specification remain fixed, self-improvement consists solely of updating these task-relevant components across iterations.

**Agent Skill.** Agent skill forms a persistent memory of what SAGE has learned about a formulation space (Li et al., 2026; Vishe et al., 2026). It records interaction-level failures, successful counterexamples, preferred component families, categorical concentration regimes, and effective search operators. The LLM reflector cannot modify numerical concentration bounds, which are generated and validated exclusively by deterministic code. Agent skill reshapes the composer's search prior to avoid known failures while encoding task-specific local design rules rather than broad blacklists.

**Task Prompt.** The task prompt is the controlled context given to an agent at the start of the next iteration (Wang et al., 2024; Zhou et al., 2026). It states the current gate of interest, the physical failure that motivates that gate, the relevant evidence, and the expected reasoning for that role (Pei et al., 2025). Prompt updates matter when the model must reinterpret the same task in light of new information, for example by weighing a mobility failure against evidence of ion association. They change how the agent reasons without introducing numerical values or shifting the pass criteria.

**Tool-Use Strategy.** The tool-use strategy determines how the workflow directs its subsequent retrieval and search actions across scientific tools (Ruan et al., 2023). It can prioritize retrieval for a failed gate, shift attention to different evidence sources, or favor operators that probe revised interactions. This component is important because it redirects tool use toward more informative, task-relevant actions and helps prevent repeated generic or unsuccessful search behavior.

The reflector routes each actionable failure to the part of the system best suited to address it. Missing evidence mainly leads to tool-use updates, because the fix is to gather better information. Composition or compatibility failures create interaction-aware skills and adjust future search preferences. A physical-threshold miss can update the skills and retrieval focus while keeping the threshold itself unchanged, so that search improves without weakening the criterion. A recoverable process error may update the prompts for the affected role, but is excluded from scientific failures so that engineering issues are not confused with chemical lessons. Let $\Delta\pi_{t,i} = (\Delta s_{t,i}, \Delta p_{t,i}, \Delta u_{t,i})$ denote the proposed changes to skills, prompts, and tool-use after iteration $i$. An update is applied only through

$$\pi_{t,i+1} = \begin{cases} \mathrm{Commit}(\pi_{t,i}, \Delta\pi_{t,i}), & \mathrm{Guard}(\Delta\pi_{t,i}) = 1, \\ \pi_{t,i}, & \text{otherwise.} \end{cases} \tag{5}$$

In Equation (5), the guard validates each proposed policy update against the current task specification and failure history. It accepts a change only if the update addresses a resolvable failure, targets a valid component, leaves the task specification and gate set unchanged, and remains compatible with the downstream search interface. Once committed, $\pi_{t,i+1}$ becomes the design policy used by the next call to Equation (3), and it can never alter the physical evaluation of a formulation.

Table 1: Designed formulations for high-voltage electrolytes; see Section A.5 for additional examples.

| ID | Salt Concentration (mol/L) | Solvent Molar Ratio | Additive (mol) |
|---|---|---|---|
| 01 | $LiBF_4$ 1.04; $LiPF_6$ 0.69 | GBL 32.0; METHF 28.3; DEE 16.2; DMC 23.4 | DTD 3.00 |
| 02 | $LiPF_6$ 1.11; $LiBF_4$ 0.44 | THF 1.1; FEC 27.2; METHF 48.4; AN 23.3 | VC 3.00 |
| 03 | $LiPF_6$ 1.29; $LiBF_4$ 0.44 | SL 1.1; FEC 27.2; METHF 48.4; AN 23.3 | PS 3.00 |

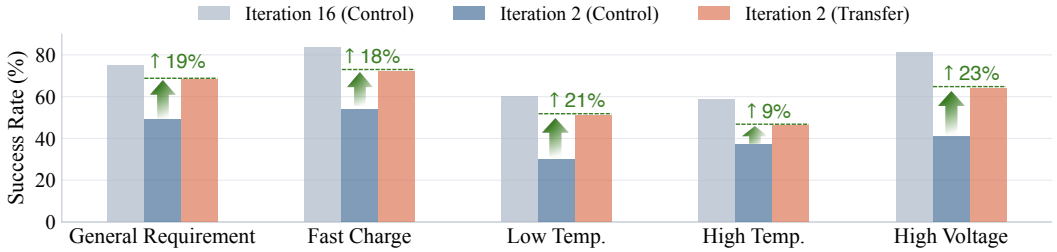



Figure 6: Transferring structured artifacts distilled by SAGE improves performance in fresh runs of the corresponding tasks using the DeepSeek backbone. Additional results are provided in Section A.7.

SAGE further protects this process with a conservative rollback rule that undoes an update if it harms the search. Building on Equation (2), we define the discovery gain $\Delta V_{t,i} = |\mathcal{V}_{t,i} \setminus \mathcal{V}_{t,i-1}|$ as the number of newly discovered valid formulations, which is distinct from the candidate-level violation score $V_t(c)$. Let $K_{t,i} \neq \emptyset$ denote the set of concrete, task-specific failure modes named by the update committed after iteration $i$, which is distinct from the evaluated candidate subset $\mathcal{K}_{t,i}$. Let $f_k(i)$ denote the fraction of physically submitted candidates in iteration $i$ that exhibit mode $k$, excluding execution-only failures from the numerator. If no comparable candidates are submitted, no rollback is triggered. After observing iteration $i+1$, the update from Equation (5) is reverted only when

$$\Delta V_{t,i+1} < \Delta V_{t,i} \quad \text{and} \quad \forall k \in K_{t,i},\ f_k(i+1) \geq f_k(i). \tag{6}$$

Intuitively, the first condition in Equation (6) captures a drop in discovery gain, and the second requires that none of the targeted failure modes have improved. When a rollback occurs, the reverted update and the reason for its removal remain in the task history, so later iterations do not repeat the same mistake. As a result, this safeguard limits harmful policy drift through a rigorous boundary.

As presented in Figure 1, SAGE forms a closed discovery loop in which evidence and physical evaluation iteratively refine the search under fixed constraints. Evidence and task history generate failure-specific hypotheses through Equation (3), which deterministic programs then translate into feasible and diverse candidate formulations. MD and DFT then provide objective feedback, with Equation (4) quantifying uncertainty and Equation (2) admitting only candidates that satisfy all required gates. Verified failures update skills, prompts, and tool-use strategies through Equation (5), improving subsequent iterations, while Equation (6) prevents unsupported policy changes and preserves an auditable discovery trajectory. In this way, SAGE progressively improves its search strategy and narrows the design space to identify formulations that satisfy physicochemical constraints. Implementation details are provided in Section A.3. We also present a concrete case study in Section A.10 to illustrate how design policies, hypotheses, and supporting evidence evolve in response to constraint failures in high-voltage electrolyte formulation design.

# 3 Experiments

## 3.1 Experimental Setup

We evaluate SAGE on five industrial electrolyte formulation tasks: general requirements, fast charge, low temperature, high temperature, and high voltage. These tasks impose distinct composition, transport, and stability constraints while sharing common liquid-state and quality-control requirements. All baselines adopt the same admissibility checks, deterministic programs, scientific tools, and evaluation protocol, differing only in retrieval, reasoning, and candidate proposal. The primary metric is canonical PASS, which requires passing all gates, and we report PASS rates under fixed computational budgets for all baselines. Moreover, we compare SAGE with single-LLM and tool-augmented

Table 2: Main results on electrolyte formulation design. PASS counts are reported out of 80 candidates per task under an identical protocol (*e.g.,* the same budget, evaluation scheme, and database access).

| Method | | Task 1 | Task 2 | Task 3 | Task 4 | Task 5 | Avg. Rate |
|---|---|---|---|---|---|---|---|
| *Backbone* | *Agentic Design* | *General* | *Fast Charge* | *Low Temp.* | *High Temp.* | *High Volt.* | |
| *Large Language Models (LLMs)* | | | | | | | |
| GPT-5.6 Terra | LLM + web search | 27 | 22 | 10 | 18 | 5 | 20.5% |
| LLaMat-3 | LLM + web search | 13 | 6 | 0 | 0 | 1 | 5.0% |
| NatureLM 8-7B | LLM + web search | 14 | 15 | 5 | 9 | 0 | 10.8% |
| DeepSeek V4 Flash | LLM + web search | 17 | 16 | 11 | 17 | 0 | 15.3% |
| Qwen3.8-27B | LLM + web search | 19 | 5 | 3 | 0 | 1 | 7.0% |
| *General-Purpose Agentic Workflow* | | | | | | | |
| GPT-5.6 Terra | Agent with tool use | 31 | 15 | 13 | 20 | 4 | 20.8% |
| LLaMat-3 | Agent with tool use | 17 | 4 | 1 | 2 | 2 | 6.5% |
| NatureLM 8-7B | Agent with tool use | 20 | 14 | 6 | 8 | 0 | 12.0% |
| DeepSeek V4 Flash | Agent with tool use | 28 | 21 | 12 | 19 | 4 | 21.0% |
| Qwen3.8-27B | Agent with tool use | 12 | 15 | 3 | 2 | 0 | 8.0% |
| *Adapted Agentic Workflows* | | | | | | | |
| ChemCrow | ReAct Controller | 31 | 17 | 8 | 14 | 0 | 17.5% |
| ChemGraph | Planner Worker | 22 | 21 | 9 | 15 | 3 | 17.5% |
| MatSciAgent | Router Specialist | 24 | 8 | 7 | 10 | 3 | 13.0% |
| DeepSeek V4 Flash | **SAGE (Ours)** | **60** | **67** | **48** | **47** | **65** | **71.8%** |

single-agent baselines built on GPT-5.6 Terra (Singh et al., 2025), DeepSeek V4 Flash (Xu et al., 2026), LLaMat-3 (Mishra et al., 2024), NatureLM 8-7B (Xia et al., 2025), and Qwen3.8-27B (Yang et al., 2025), and scientifically adapted agentic workflows including ChemCrow (M. Bran et al., 2024), ChemGraph (Pham et al., 2026), and MatSciAgent (Chaudhari et al., 2026). We also leverage public resources such as Bamboo-Mixer (Yang et al., 2026) and CALiSol-23 (de Blasio et al., 2024) for evidence retrieval, candidate-space construction, and novelty checking. In all comparisons, candidates are evaluated under the same task specifications and canonical decision logic, with final physical labels determined solely by the fixed evaluation protocols. Notably, Figures 3 to 5 and 7 visualize the results for task 1. Detailed experimental settings are provided in Section A.4.

### 3.2 Main Results

Task 1 requires general-purpose, compatibility-checked formulations to satisfy conductivity under shared density, coordination, and viscosity limits, while task 2 adds stricter transport requirements for fast charging. Task 3 and task 4 apply similar gates at $-20^{\circ}\mathrm{C}$ and $55^{\circ}\mathrm{C}$, with task 4 additionally constraining coordination drift across temperatures. Task 5 further adds a DFT-verified relative cluster vertical ionization potential requirement. All tasks share the same quality-control criteria and frozen thresholds, and each method receives a fixed budget of 80 candidates per task. Baselines include fixed-prompt LLMs, tool-using single agents on comparable backbones, and adapted agentic workflows evaluated under the same audited tool and gate interface as SAGE, enabling controlled comparison at the system level. As shown in Table 2, SAGE achieves an average PASS rate of $71.8\%$, more than three times the strongest baseline, with the largest gains on the most constrained tasks. Comparisons further show that the improvement comes from SAGE's closed-loop use of physical feedback to refine hypotheses and guide search toward compliant regions, rather than from the base model alone. In addition, Table 1 presents three representative high-voltage electrolyte formulations designed by SAGE, with further examples and details in Section A.5.

The performance gap arises because electrolyte formulation is a coupled optimization problem. Raising salt concentration can improve carrier density but also increase clustering and viscosity (Suo et al., 2015). Weak-donor co-solvents can lower melting points but destabilize coordination structure (Park et al., 2023). Wide-window co-solvents can improve oxidative stability but often reduce mobility or interphase-forming ability (Zhang et al., 2025). As a result, frequently designed single-salt or single-solvent recipes often satisfy one objective while failing others once all physical gates are enforced together. This helps explain why fixed-prompt and simple tool-using baselines, which rely on high-frequency priors, plateau at low PASS rates regardless of backbone strength. In contrast, SAGE's accepted formulations exhibit explicit compensatory design. In task 2, dual-salt systems are paired with fluorinated diluents to offset viscosity penalties while preserving additive effects. In task 3, low-melting ether and ester co-solvents are explored in mixed-salt recipes under the $-20^{\circ}\mathrm{C}$ screen. In task 5, sulfone or nitrile co-solvents may be combined with CEI-relevant additives as a design strategy, while successful candidates satisfy the bulk screens and relative cluster vertical ionization

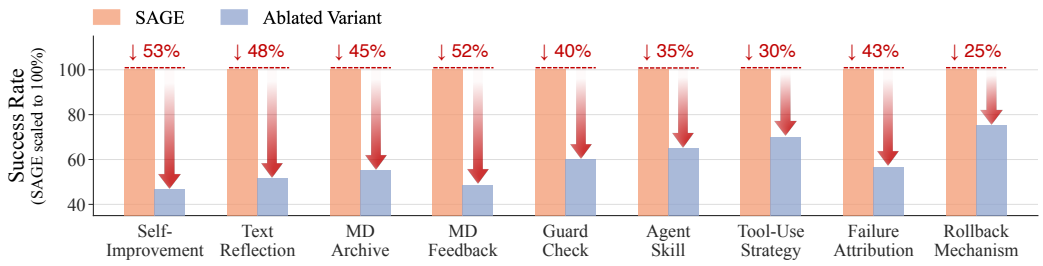



Figure 7: **Ablations on SAGE**. Removing any core component significantly degrades performance, highlighting the importance of self-improvement, physics-driven feedback, and failure-aware learning.

potential (VIP) criterion. These patterns suggest that SAGE performs better not by searching more broadly in an unguided way, but by using failure signals to identify which constraint is binding and then introducing targeted compensatory changes in other components. This also illustrates why SAGE discovers more complex formulations: once improvement is driven by cross-constraint correction rather than one-shot heuristic guessing, the search naturally expands toward multi-component recipes that coordinate several functional roles at once. Consistent with this, SAGE's high novelty share against the indexed database shows that it is not simply recovering known valid recipes, but validating new regions of the coupled composition–property space. We offer seven additional categories of main experiments and ablation studies in Sections A.6 and A.7, along with visualizations in Section A.8.

### 3.3 Ablation Studies

We conduct ablation studies to identify the source of SAGE's gains. As shown in Figure 5, swapping the backbone across multiple LLMs of different sizes shows that SAGE consistently improves success rates by 40 to 54 points over the corresponding base model, confirming that the advantage comes from our agentic loop rather than any single backbone. Figure 7 shows that removing or replacing individual components from the full workflow consistently lowers success rates, with the largest drops occurring when self-improvement or MD feedback is removed or when structured policy updates are replaced by text-only reflection. This suggests that SAGE's gains mainly stem from closing the loop between physical evaluation outcomes and the agent's self-evolving hypotheses about viable composition trade-offs, rather than from broader search or better tool orchestration alone. Smaller but still meaningful declines from guard checks, rollback, and failure attribution further show that these components help steer search away from repeatedly infeasible regions. Taken together, the ablations suggest that SAGE's advantage comes not from the backbone model or any single dominant module, but from the strategic combination of physics-grounded feedback and iterative self-improvement.

Figure 6 shows that structured knowledge (*i.e.,* design policy update logs; detailed in Section A.9) accumulated over iterations in prior runs of the same task can be reused to improve performance when the task is revisited, with gains emerging as early as iteration 2. This transfer closes a substantial portion of the gap between the cold-start iteration-2 baseline and the fully self-improved iteration-16 result, suggesting that much of the benefit of long-horizon exploration can be distilled into reusable artifacts rather than relying solely on online trial-and-error within a single run. The smaller gain under high-temperature conditions suggests that shared design trade-offs transfer well, while more task-specific constraints still require in-domain refinement. Overall, these results show that SAGE can preserve reusable design knowledge to accelerate future exploration, which is a practical advantage in recurring industrial formulation tasks. Additional ablations are provided in Section A.7.

## 4 Conclusion

In this work, we present SAGE, a self-improving, physics-driven agentic framework for designing electrolyte formulations under complex physicochemical constraints. Specifically, SAGE learns from failures to iteratively refine its design policy, updating its hypotheses, search strategy, and candidate generation process via physics-grounded feedback, thereby forming a scientifically auditable and rigorous discovery loop. Empirical results demonstrate its promise for advancing real-world electrolyte development in industrial battery R&D, while ablation studies validate its key design choices.

## REFERENCES


Jiaru Bai, Sebastian Mosbach, Connor J Taylor, Dogancan Karan, Kok Foong Lee, Simon D Rihm, Jethro Akroyd, Alexei A Lapkin, and Markus Kraft. A Dynamic Knowledge Graph Approach to Distributed Self-Driving Laboratories. *Nature Communications*, 15(1):462, 2024. 16

Anass Benayad, Diddo Diddens, Andreas Heuer, Anand Narayanan Krishnamoorthy, Moumita Maiti, Frédéric Le Cras, Maxime Legallais, Fuzhan Rahmanian, Yuyoung Shin, Helge Stein, et al. High-Throughput Experimentation and Computational Freeway Lanes for Accelerated Battery Electrolyte and Interface Development Research. *Advanced Energy Materials*, 12(17):2102678, 2022. 16

JA Bennett, N Orouji, M Khan, S Sadeghi, J Rodgers, and M Abolhasani. Autonomous Reaction Pareto-Front Mapping with A Self-Driving Catalysis Laboratory. *Nature Chemical Engineering*, 1 (3):240–250, 2024. 16

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous Chemical Research with Large Language Models. *Nature*, 624(7992):570–578, 2023. 16

Gabriel Bradford, Jeffrey Lopez, Jurgis Ruza, Michael A Stolberg, Richard Osterude, Jeremiah A Johnson, Rafael Gomez-Bombarelli, and Yang Shao-Horn. Chemistry-Informed Machine Learning for Polymer Electrolyte Discovery. *ACS Central Science*, 9(2):206, 2023. 16

Akshat Chaudhari, Janghoon Ock, and Amir Barati Farimani. Modular Large Language Model Agents for Multi-Task Computational Materials Science. *Communications Materials*, 7(1):131, 2026. 8, 20

Tianwei Dai, Sriram Vijayakrishnan, Filip T Szczypiński, Jean-François Ayme, Ehsan Simaei, Thomas Fellowes, Rob Clowes, Lyubomir Kotopanov, Caitlin E Shields, Zhengxue Zhou, et al. Autonomous Mobile Robots for Exploratory Synthetic Chemistry. *Nature*, 635(8040):890–897, 2024. 16

Adarsh Dave, Jared Mitchell, Sven Burke, Hongyi Lin, Jay Whitacre, and Venkatasubramanian Viswanathan. Autonomous Optimization of Non-Aqueous Li-Ion Battery Electrolytes via Robotic Experimentation and Machine Learning Coupling. *Nature Communications*, 13(1):5454, 2022. 2, 16

Paolo de Blasio, Jonas Elsborg, Tejs Vegge, Eibar Flores, and Arghya Bhowmik. CALiSol-23: Experimental Electrolyte Conductivity Data for Various Li-Salts and Solvent Combinations. *Scientific Data*, 11(1):750, 2024. 8, 20

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2): 182–197, 2002. 18

Alireza Ghafarollahi and Markus J Buehler. SciAgents: Automating Scientific Discovery Through Bioinspired Multi-Agent Intelligent Graph Reasoning. *Advanced Materials*, 37(22):2413523, 2025. 16

Guinevere A Giffin. The Role of Concentration in Electrolyte Solutions for Non-Aqueous Lithium-Based Batteries. *Nature Communications*, 13(1):5250, 2022. 1, 19

Clare P Grey and David S Hall. Prospects for Lithium-Ion Batteries and Beyond—A 2030 Vision. *Nature Communications*, 11(1):6279, 2020. 1

Vinamr Jain, Zhilong Wang, and Fengqi You. Machine Learning Pipelines for the Design of Solid-State Electrolytes. *Materials Horizons*, 13(1):15–44, 2026. 2

William L Jorgensen, David S Maxwell, and Julian Tirado-Rives. Development and Testing of the OPLS All-Atom Force Field on Conformational Energetics and Properties of Organic Liquids. *Journal of the American Chemical Society*, 118(45):11225–11236, 1996. 5, 32, 33

Jaemin Kim, Ke-Hsin Wang, Ritesh Kumar, Peiyuan Ma, and Chibueze V Amanchukwu. Generative Electrolyte Solvent and Formulation Discovery. *JACS Au*, 6(4):2288, 2026. 16, 20

Sang Cheol Kim, Solomon T Oyakhire, Constantine Athanitis, Jingyang Wang, Zewen Zhang, Wenbo Zhang, David T Boyle, Mun Sek Kim, Zhiao Yu, Xin Gao, et al. Data-Driven Electrolyte Design for Lithium Metal Anodes. *Proceedings of the National Academy of Sciences*, 120(10):e2214357120, 2023. 16

Xiangyi Li, Yimin Liu, Wenbo Chen, Bingran You, Zonglin Di, Yifeng He, Shenghan Zheng, Kyoung Whan Choe, Jiankai Sun, Shuyi Wang, et al. SkillsBench: Benchmarking How Well Agent Skills Work Across Diverse Tasks. *arXiv preprint arXiv:2602.12670*, 2026. 6

Changlin Liu, Lizhi Sheng, and Lili Jiang. Research on Performance Constraints and Electrolyte Optimization Strategies for Lithium-Ion Batteries at Low Temperatures. *RSC Advances*, 15(10): 7995–8018, 2025. 2, 20

Qianqian Liu and Liguang Wang. Fundamentals of Electrolyte Design for Wide-Temperature Lithium Metal Batteries. *Advanced Energy Materials*, 13(37):2301742, 2023. 20

ER Logan and JR Dahn. Electrolyte Design for Fast-Charging Li-Ion Batteries. *Trends in Chemistry*, 2(4):354–366, 2020. 19

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery. *arXiv preprint arXiv:2408.06292*, 2024. 16

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting Large Language Models with Chemistry Tools. *Nature Machine Intelligence*, 6(5):525–535, 2024. 8, 16, 20

Vaibhav Mishra, Somaditya Singh, Dhruv Ahlawat, Mohd Zaki, Vaibhav Bihani, Hargun Singh Grover, Biswajit Mishra, Santiago Miret, NM Krishnan, et al. Foundational Large Language Models for Materials Research. *arXiv preprint arXiv:2412.09560*, 2024. 8, 20

Aashutosh Mistry, Zhou Yu, Brandon L Peters, Chao Fang, Rui Wang, Larry A Curtiss, Nitash P Balsara, Lei Cheng, and Venkat Srinivasan. Toward Bottom-Up Understanding of Transport in Concentrated Battery Electrolytes. *ACS Central Science*, 8(7):880, 2022. 2, 19, 24

Juran Noh, Hieu A Doan, Heather Job, Lily A Robertson, Lu Zhang, Rajeev S Assary, Karl Mueller, Vijayakumar Murugesan, and Yangang Liang. An Integrated High-Throughput Robotic Platform and Active Learning Approach for Accelerated Discovery of Optimal Electrolyte Formulations. *Nature Communications*, 15(1):2757, 2024. 2, 16

Sewon Park, Saehun Kim, Jeong-A Lee, Makoto Ue, and Nam-Soon Choi. Liquid Electrolyte Chemistries for Solid Electrolyte Interphase Construction on Silicon and Lithium-Metal Anodes. *Chemical Science*, 14(37):9996–10024, 2023. 1, 8, 25

Zehua Pei, Hui-Ling Zhen, Shixiong Kai, Sinno Jialin Pan, Yunhe Wang, Mingxuan Yuan, and Bei Yu. Scope: Prompt Evolution for Enhancing Agent Effectiveness. *arXiv preprint arXiv:2512.15374*, 2025. 6

Zehang Peng, Kui Ding, Meiting Lai, Rui Qiu, Ye Xiao, Junkai Shi, Xiaoxian Guan, Yue-Peng Cai, Chao Xu, Fei Wang, et al. Rational Electrolyte Solvent Screening for High-Energy Lithium Metal Batteries at Low Temperatures. *Nature Communications*, 17(1):590, 2025. 20

Thang D Pham, Aditya Tanikanti, and Murat Keçeli. ChemGraph as An Agentic Framework for Computational Chemistry Workflows. *Communications Chemistry*, 9(1):33, 2026. 8, 20

Kaiqiang Qin, Kathryn Holguin, Motahareh Mohammadiroudbari, Jinghao Huang, Eric Young Sam Kim, Rosemary Hall, and Chao Luo. Strategies in structure and Electrolyte Design for High-Performance Lithium Metal Batteries. *Advanced Functional Materials*, 31(15):2009694, 2021. 1

Jacob T Rapp, Bennett J Bremer, and Philip A Romero. Self-Driving Laboratories to Autonomously Navigate the Protein Fitness Landscape. *Nature Chemical Engineering*, 1(1):97–107, 2024. 16

Xiaodi Ren, Lianfeng Zou, Xia Cao, Mark H Engelhard, Wen Liu, Sarah D Burton, Hongkyung Lee, Chaojiang Niu, Bethany E Matthews, Zihua Zhu, et al. Enabling High-Voltage Lithium-Metal Batteries Under Practical Conditions. *Joule*, 3(7):1662–1676, 2019a. 1, 20

Xiaodi Ren, Lianfeng Zou, Shuhong Jiao, Donghai Mei, Mark H Engelhard, Qiuyan Li, Hongkyung Lee, Chaojiang Niu, Brian D Adams, Chongmin Wang, et al. High-Concentration Ether Electrolytes for Stable High-Voltage Lithium Metal Batteries. *ACS Energy Letters*, 4(4):896–902, 2019b. 1

Jingqing Ruan, YiHong Chen, Bin Zhang, Zhiwei Xu, Tianpeng Bao, du qing, shi shiwei, Hangyu Mao, Xingyu Zeng, and Rui Zhao. TPTU: Task Planning and Tool Usage of Large Language Model-based AI Agents. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023. URL https://openreview.net/forum?id=GrkgKtOjaH. 6

Yixiang Ruan, Chenyin Lu, Ning Xu, Yuchen He, Yixin Chen, Jian Zhang, Jun Xuan, Jianzhang Pan, Qun Fang, Hanyu Gao, et al. An Automatic End-to-End Chemical Synthesis Development Platform Powered by Large Language Models. *Nature Communications*, 15(1):10160, 2024. 16

Jurģis Ruža and Rafael Gomez-Bombarelli. Reasoning-to-Simulation: An Agentic Framework for Discovery of Electrolyte Materials. In *AI for Accelerated Materials Design - ICLR 2026*, 2026. URL https://openreview.net/forum?id=MEXl18VhBL. 4, 16

Vidushi Sharma, Maxwell Giammona, Dmitry Zubarev, Andy Tek, Khanh Nugyuen, Linda Sundberg, Daniele Congiu, and Young-Hye La. Formulation Graphs for Mapping Structure-Composition of Battery Electrolytes to Device Performance. *Journal of Chemical Information and Modeling*, 63 (22):6998, 2023. 1, 2, 16

Malcolm Sim, Mohammad Ghazi Vakili, Felix Strieth-Kalthoff, Han Hao, Riley J Hickman, Santiago Miret, Sergio Pablo-García, and Alán Aspuru-Guzik. ChemOS 2.0: An Orchestration Architecture for Chemical Self-Driving Laboratories. *Matter*, 7(9):2959–2977, 2024. 16

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025. 8, 20, 33

Yuanyuan Song, Jiazhe Ju, Jifeng Wang, Kai Li, Xia Wang, Rui Gao, Hongbin Lu, Dongliang Chao, and Ying Wang. Multi-Objective Optimization of Ionic Polymer Electrolytes for High-Voltage Fast-Charging and Versatile Lithium Batteries. *Advanced Materials*, 37(17):2500941, 2025. 1, 2, 19, 25

Haoyang Su, Renqi Chen, Shixiang Tang, Zhenfei Yin, Xinzhe Zheng, Jinzhe Li, Biqing Qi, Qi Wu, Hui Li, Wanli Ouyang, et al. Many Heads Are Better Than One: Improved Scientific Idea Generation by A LLM-Based Multi-Agent System. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28201–28240, 2025. 16

Liumin Suo, Oleg Borodin, Tao Gao, Marco Olguin, Janet Ho, Xiulin Fan, Chao Luo, Chunsheng Wang, and Kang Xu. "Water-in-Salt" Electrolyte Enables High-Voltage Aqueous Lithium-Ion Chemistries. *Science*, 350(6263):938–943, 2015. 1, 8

Nathan J Szymanski, Pragnay Nevatia, Christopher J Bartel, Yan Zeng, and Gerbrand Ceder. Autonomous and Dynamic Precursor Selection for Solid-State Materials Synthesis. *Nature Communications*, 14(1):6956, 2023a. 16

Nathan J Szymanski, Bernardus Rendy, Yuxing Fei, Rishi E Kumar, Tanjin He, David Milsted, Matthew J McDermott, Max Gallant, Ekin Dogus Cubuk, Amil Merchant, et al. An Autonomous Laboratory for the Accelerated Synthesis of Inorganic Materials. *Nature*, 624(7990):86–91, 2023b. 16

Yash Vishe, Rohan Surana, Xunyi Jiang, Zihan Huang, Xintong Li, Nikki Lijing Kuang, Tong Yu, Ryan A Rossi, Jingbo Shang, Julian McAuley, et al. Skill-R1: Agent Skill Evolution via Reinforcement Learning. *arXiv preprint arXiv:2605.09359*, 2026. 6

Amanda A Volk, Robert W Epps, Daniel T Yonemoto, Benjamin S Masters, Felix N Castellano, Kristofer G Reyes, and Milad Abolhasani. AlphaFlow: Autonomous Discovery and Optimization of Multi-Step Chemistry Using A Self-Driven Fluidic Lab Guided by Reinforcement Learning. *Nature Communications*, 14(1):1403, 2023. 16

Hongli Wan, Jijian Xu, and Chunsheng Wang. Designing Electrolytes and Interphases for High-Energy Lithium Batteries. *Nature Reviews Chemistry*, 8(1):30–44, 2024. 1, 20, 25

Aiping Wang, Sanket Kadam, Hong Li, Siqi Shi, and Yue Qi. Review on Modeling of the Anode Solid Electrolyte Interphase (SEI) for Lithium-Ion Batteries. *NPJ Computational Materials*, 4(1): 15, 2018. 1

Bingning Wang, Hieu A Doan, Seoung-Bum Son, Daniel P Abraham, Stephen E Trask, Andrew Jansen, Kang Xu, and Chen Liao. Data-Driven Design of Electrolyte Additives Supporting High-Performance 5 V LiNi0. 5Mn1. 5O4 Positive Electrodes. *Nature Communications*, 16(1):3413, 2025a. 2

Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric Xing, and Zhiting Hu. PromptAgent: Strategic Planning with Language Models Enables Expert-level Prompt Optimization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=22pyNMuIoa. 6

Zhi-jie Wang, BJEM Zhang, et al. Weakly Solvating Electrolytes for Next-Generation Lithium Batteries: Design Principles and Recent Advances. *Energy Materials and Devices*, 1(1), 2023. 1, 24

Zhilong Wang and Fengqi You. A Dynamic Routing-Guided Interpretable Framework for Salt–Solvent Chemistry. *Nature Computational Science*, 6(3):271–284, 2026. 20

Zhilong Wang, Wolfgang G Zeier, and Fengqi You. Toward AI Ecosystems for Electrolyte and Interface Engineering in Solid-State Batteries. *Science Advances*, 11(48):eaea0638, 2025b. 16

Jiaqi Wei, Yuejin Yang, Xiang Zhang, Yuhan Chen, Xiang Zhuang, Zhangyang Gao, Dongzhan Zhou, Guangshuai Wang, Zhiqiang Gao, Juntai Cao, et al. From AI for Science to Agentic Science: A survey on Autonomous Scientific Discovery. *arXiv preprint arXiv:2508.14111*, 2025. 16

Shuixin Xia, Xinsheng Wu, Zhichu Zhang, Yi Cui, and Wei Liu. Practical Challenges and Future Perspectives of All-Solid-State Lithium-Metal Batteries. *Chem*, 5(4):753–785, 2019. 1

Yingce Xia, Peiran Jin, Shufang Xie, Liang He, Chuan Cao, Renqian Luo, Guoqing Liu, Yue Wang, Zequn Liu, Yuan-Jyue Chen, et al. Nature Language Model: Deciphering the Language of Nature for Scientific Discovery. *arXiv preprint arXiv:2502.07527*, 2025. 8, 20

Jie Xiao, Nicole Adelstein, Yujing Bi, Wenjuan Bian, Jordi Cabana, Corie L Cobb, Yi Cui, Shen J Dillon, Marca M Doeff, Saiful M Islam, et al. Assessing Cathode–Electrolyte Interphases in Batteries. *Nature Energy*, 9(12):1463–1473, 2024. 1, 20

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-V4: Towards Highly Efficient Million-Token Context Intelligence. *arXiv preprint arXiv:2606.19348*, 2026. 8, 17, 20

Yizhe Yan, Feng Hai, Bin Wang, Wenrui Cao, Mingtao Li, Chaohui Wang, Naipeng Li, and Dan Zhao. Machine Learning Accelerates High-Voltage Electrolyte Discovery for Lithium Metal Batteries. *Energy Storage Materials*, 79:104312, 2025. 2, 16

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025. 8, 20, 33

Huicong Yang, Juan Li, Zhenhua Sun, Ruopian Fang, Da-Wei Wang, Kuang He, Hui-Ming Cheng, and Feng Li. Reliable Liquid Electrolytes for Lithium Metal Batteries. *Energy Storage Materials*, 30:113–129, 2020. 1

Zhenze Yang, Yifan Wu, Xu Han, Ziqing Zhang, Haoen Lai, Zhenliang Mu, Tianze Zheng, Siyuan Liu, Zhichen Pu, Zhi Wang, et al. A Unified Predictive and Generative Solution for Liquid Electrolyte Formulation. *Nature Machine Intelligence*, 8(2):186–196, 2026. 2, 8, 16, 20

Jackie T Yik, Leiting Zhang, Jens Sjölund, Xu Hou, Per H Svensson, Kristina Edström, and Erik J Berg. Automated Electrolyte Formulation and Coin Cell Assembly for High-Throughput Lithium-Ion Battery Research. *Digital Discovery*, 2(3):799–808, 2023. 16

Pengbo Zhai, Lixuan Liu, Xiaokang Gu, Tianshuai Wang, and Yongji Gong. Interface Engineering for Lithium Metal Anodes in Liquid Electrolyte. *Advanced Energy Materials*, 10(34):2001257, 2020. 1

Hengrui Zhang, Jie Chen, James M Rondinelli, and Wei Chen. Molsets: Molecular Graph Deep Sets Learning for Mixture Property Modeling. *arXiv preprint arXiv:2312.16473*, 2023. 20

Weili Zhang, Yang Lu, Qingqing Feng, Hao Wang, Guangyu Cheng, Hao Liu, Qingbin Cao, Zhenjun Luo, Pan Zhou, Yingchun Xia, et al. Multifunctional Electrolyte Additive for High Power Lithium Metal Batteries at Ultra-Low Temperatures. *Nature Communications*, 16(1):3344, 2025. 8

Yu Zhang, Yan Lu, Jun Jin, Meifen Wu, Huihui Yuan, Shilin Zhang, Kenneth Davey, Zaiping Guo, and Zhaoyin Wen. Electrolyte Design for Lithium-Ion Batteries for Extreme Temperature Applications. *Advanced Materials*, 36(13):2308484, 2024. 25

Han Zhou, Xingchen Wan, Ruoxi Sun, Hamid Palangi, Shariq Iqbal, Ivan Vulić, Anna Korhonen, and Sercan O Arik. Multi-Agent Design: Optimizing Agents with Better Prompts and Topologies. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=I05H9RUzHB. 6

Murtaza Zohair, Vidushi Sharma, Eduardo A Soares, Khanh Nguyen, Maxwell Giammona, Linda Sundberg, Andy Tek, Emilio Vital Brazil, and Young-Hye La. Chemical Foundation Model-Guided Design of High Ionic Conductivity Electrolyte Formulations. *npj Computational Materials*, 11(1): 283, 2025. 2, 16

# CONTENTS

**1 Introduction** **1**

**2 Method** **2**

2.1 Problem Setup . . . 3

2.2 Evidence-Aware Agentic Search Workflow . . . 4

2.3 Physics-Driven Candidate Evaluation . . . 5

2.4 Self-Improvement for Design Policy Refinement . . . 6

**3 Experiments** **7**

3.1 Experimental Setup . . . 7

3.2 Main Results . . . 8

3.3 Ablation Studies . . . 9

**4 Conclusion** **9**

**A Appendix** **16**

A.1 AI Use Statement . . . 16

A.2 Related Work . . . 16

A.3 Implementation Details . . . 17

A.4 Detailed Experimental Setup . . . 19

A.5 Representative Formulation Examples . . . 23

A.6 Evaluation of Formulation Novelty . . . 28

A.7 Additional Main Results and Ablations . . . 29

A.8 3D Visualization of Designed Formulations . . . 36

A.9 Artifacts from the Final Iteration Improve Task Performance . . . 37

A.10 Case Study: High-Voltage Battery Formulation Design . . . 40

A.11 Limitations and Future Work . . . 53

# A Appendix

## A.1 AI Use Statement

In this work, we used generative AI tools to (1) improve the clarity and coherence of the writing; (2) retrieve background knowledge and factual information relevant to the content, such as related papers and domain-specific terminology; and (3) assist in debugging the code.

## A.2 Related Work

**Agentic Science**. Recent work in agentic science has moved AI for scientific discovery beyond one-shot prediction toward workflows that coordinate planning, tool use, interaction, and evaluation across multiple steps (Wei et al., 2025). Language-model-based scientific assistants combine reasoning with internet search, code execution, and laboratory automation to orchestrate heterogeneous scientific operations (Boiko et al., 2023; M. Bran et al., 2024). This line of work shows that scientific tools can provide procedural and numerical grounding beyond language-model reasoning alone. Self-driving laboratories extend this paradigm to closed-loop experimentation by selecting experiments, synthesizing materials, and incorporating measured outcomes under predefined objectives and protocols (Szymanski et al., 2023b;a). Other platforms automate synthesis development, exploratory chemistry, flow chemistry, and reaction optimization through active selection or reinforcement learning (Ruan et al., 2024; Dai et al., 2024; Volk et al., 2023; Bennett et al., 2024). Orchestration architectures and distributed knowledge representations further support scheduling, provenance tracking, and coordination across laboratory resources (Sim et al., 2024; Bai et al., 2024). These approaches demonstrate the practical significance of tool use, active experiment selection, and workflow coordination for scientific discovery. Open-ended scientific agents also extend the loop to idea generation, implementation, and evaluation (Lu et al., 2024; Su et al., 2025). Multi-agent scientific reasoning and iterative protein engineering studies suggest that role decomposition can organize complex discovery tasks (Ghafarollahi & Buehler, 2025; Rapp et al., 2024). However, existing approaches generally rely on a fixed task representation, stable tool interfaces, or predefined rewards, and rarely specify how failed physical constraints become persistent and reversible updates to evidence retrieval, hypothesis formation, and candidate search. This distinction matters because a plausible proposal does not guarantee that coupled scientific constraints are satisfied, and a failed experiment does not automatically translate into a reusable design rule. Therefore, SAGE focuses on an auditable discovery loop that separates scientific reasoning from deterministic composition and physical evaluation, while using verified task-specific failures to update subsequent search strategy.

**Electrolyte Formulation Design**. Electrolyte discovery has benefited from high-throughput experimentation, robotic platforms, active learning, and computational workflows that accelerate screening and interface development (Benayad et al., 2022; Dave et al., 2022; Noh et al., 2024; Yik et al., 2023). Automated formulation and coin cell assembly reduce manual effort and improve reproducibility in repeated experiments (Yik et al., 2023). Robotic experimentation coupled with machine learning methods can efficiently explore a prescribed formulation space while using measured properties to select subsequent trials, highlighting the value of rapid physical feedback for electrolyte optimization (Dave et al., 2022; Noh et al., 2024). Moreover, data-driven models provide a complementary route by learning composition–property or formulation–performance relationships from curated measurements, with applications to lithium metal anodes, polymer electrolytes, and device-level electrolyte behavior (Sharma et al., 2023; Kim et al., 2023; Bradford et al., 2023). In addition, chemical foundation models and scientific discovery pipelines extend this paradigm to target-conditioned prediction and candidate generation (Zohair et al., 2025; Yan et al., 2025). Unified predictive–generative models and solvent representation methods further reduce the computational cost of proposing new mixture candidates (Yang et al., 2026; Kim et al., 2026). Broader AI ecosystem studies argue for connecting models with databases, simulation, and interface engineering rather than treating formulation prediction as an isolated prediction problem (Wang et al., 2025b; Ruža & Gomez-Bombarelli, 2026). Nevertheless, these approaches optimize a static objective over a fixed

dataset, candidate library, or predefined experimental design and do not preserve a structured archive of failed formulations. They also rarely use specific physical failures to revise the next retrieval and composition policy or distinguish execution failures from chemical threshold failures. This limitation is vital in formulation design, where changing component identity or concentration can affect several coupled electrochemical properties while simulation and experimental budgets remain limited. As a result, a more rigorous and complete discovery pipeline must use physically grounded evaluation both as an acceptance mechanism and as a source of structured, auditable updates to the search policy.

### A.3 Implementation Details

The SAGE workflow is built on a deliberate limitation of the language model's authority. An LLM that directly generates a final formulation must make four distinct judgments in a single opaque step. It must interpret protocol-specific literature, propose beneficial salt–solvent interactions, assign numerical concentrations and ratios, and determine whether the candidate satisfies fixed physical criteria. Merging these judgments into one step creates several risks. The model may conflate a value reported for a related system with a prediction for the current candidate or alter an acceptance threshold in favor of its own proposal. It also becomes difficult to determine which evidence, rule, or computation produced the final claim. To avoid this, our system adopts a principle of constrained autonomy, in which each agent is limited to the form of reasoning that language models perform reliably, and every output must pass through a schema-validated artifact before it can affect any downstream step. Five agent roles–planner, retriever, proposer, evaluator, and reflector–are implemented on the same base model, DeepSeek V4 Flash (Xu et al., 2026), but differ in their role-specific prompts, output schemas, tool permissions, and the amount of open-ended reasoning required by their positions in the pipeline. Extended reasoning is enabled for the retriever, proposer, and reflector, as these modules are responsible for synthesizing evidence, generating mechanistic hypotheses, and explaining physical failures. It is disabled for the planner and evaluator, whose outputs are determined by program-level specifications, including batch size, budget, and hard constraints, that are fixed prior to invocation.

This separation is trustworthy only if every handoff between agents is auditable rather than conversational. Therefore, each stage emits a typed artifact envelope with a producer identifier, parent artifact references, and a content hash, allowing any simulated candidate to be traced to the evidence it cited, the hypothesis that produced it, the numeric generator and random seed that determined its composition, and the constraint-precheck receipt that admitted it. Tool use follows the same principle. Rather than allowing the LLM to decide when to call retrieval, search, or validation tools, the orchestrator invokes them at fixed points in the pipeline. The retriever receives the knowledge-retrieval tool, the proposer passes hypotheses to the numeric search engine, and the evaluator receives the constraint checker, novelty checker, and external validation export interface (*e.g.,* MD and DFT). Each invocation is logged as a receipt with the tool-use version, caller, input and output hashes, and execution status, so that failures can be attributed to specific tool calls. We describe this as program-mediated typed tool use rather than autonomous multi-tool orchestration because acceptance decisions for downstream physical simulation must remain reconstructible after the fact. Context budgets are also bounded by role, so context access scales with task openness rather than growing without bound across iterations.

The retrieval stage addresses a recurring limitation of static search. A query based only on the task description can surface relevant background literature, but it does not explain why a candidate failed at a particular physical gate after an unsuccessful iteration. To address this, each unresolved constraint from the previous iteration is compiled into a structured failure signature containing the gate identifier, metric, comparison direction, and relevant salt, solvent, and additive context. The retriever then expands its query set around these signatures using a task-level query, program-generated gate-specific queries, and up to three additional model-proposed queries. A second limitation is that selecting only the globally top-ranked evidence chunks can hide the few passages that directly explain a failed gate within a much larger set of broadly relevant but less useful text. We overcome this by

reserving up to four directly relevant evidence cards for each failed gate before adding more general background evidence to the context. As a result, each iteration of reasoning stays focused on the physical bottleneck identified in the previous iteration instead of repeatedly rebuilding the same background context.

A key design decision in the workflow is where numerical authority stops belonging to the language model. Letting the model output concentrations or ratios directly is problematic for three reasons: task-specific units and normalization rules are easy to misapply, language-model numbers are not deterministically reproducible from a fixed seed, and the model cannot reliably distinguish recalled values from guesses. Thus, the proposer is limited to a small set of typed hypotheses, at most eight per iteration, that specify candidate components, interaction scope, a plausible mechanism, and a suggested search operator, but never ratios, concentrations, or predicted properties. The program rejects any hypothesis that references an unregistered component, an undefined gate, or nonexistent evidence, and falls back to a low-discrepancy design-of-experiments procedure if none survive validation. A deterministic and task-specific composer then turns the valid hypotheses, together with any learned priors, into a candidate pool through constrained simplex sampling under the task's concentration and role rules. Components named by a valid hypothesis are sampled more often, and each candidate must include at least half of the components named in the hypothesis it is intended to test. This ensures that the model's suggestions shape what is generated, while all numerical values still come from the program and a fixed seed.

Once a task accumulates quality-controlled MD or DFT results, the composer shifts from evidence-conditioned sampling to a search that treats these results as a task archive rather than training data for a surrogate model. We chose this over a unified surrogate-property predictor because such a model would need to be redefined and retrained whenever a new database or protocol is introduced, and before retraining it could confidently misrank unfamiliar compositions, a harder failure mode to detect than having no ranking at all. Archive records are ranked by feasible Pareto dominance under a task-specific objective vector, with infeasible records ordered separately by normalized constraint violation. New candidates are generated by mutating the strongest parents, either by replacing a component with another admissible one of the same role or by perturbing its numerical value within a bounded multiplicative range. One operator also splits a single salt into two while keeping the total salt content fixed, allowing the effect of introducing a second salt to be tested without changing overall concentration. This is best understood as an NSGA-II-inspired real archive search rather than standard multi-generation NSGA-II (Deb et al., 2002), because every ranking signal comes from real physical feedback rather than from a surrogate-fitness landscape.

Of the twenty candidates produced in each iteration, only five can advance to physical validation, so selection is treated as a resource-allocation problem rather than a scoring problem. The search first builds a larger working pool, typically eighty hash-unique candidates, from which twenty authoritative candidates are retained after deterministic feasibility prechecks against the fixed force-field allowlist and task-level composition and compatibility constraints. These prechecks determine only whether a candidate is admissible for submission, not whether it passes any physical gate. The final five are then chosen to balance exploitation, repair, and diversity using evidence confidence, prior interaction-failure pressure, and compositional dissimilarity. The selected candidates are then frozen, together with their composition hashes and required physical gates, into a validation request that marks an explicit boundary between agentic reasoning and physical computation. Final gate outcomes are determined only after the corresponding physical evaluations are completed. This separation ensures safe retry and auditability for long-running simulations. Beyond this point, MD and, for one task, cluster-level density-functional evaluation are run under a fixed protocol, with early stopping once a gate outcome is statistically clear. Every reported quantity is therefore a meaningful measurement of transport, coordination, or oxidation behavior rather than a direct claim about experimental conductivity, cycling performance, or interfacial stability.

The subsequent part of the workflow concerns what happens after a real result returns. Failure attribution is easy to get wrong in ways that can quietly mislead later search. For example, blacklisting a component after one failed candidate ignores the fact that it may still work in a different composition. Consequently, the framework groups failures into a small set of categories, including knowledge gaps, constraint or compatibility failures, physical-threshold misses, and execution or infrastructure errors. As shown by the failure-attribution ablations in Figure 7 and Section A.7, only the first three are used for task-specific learning. Failures are stored as interaction-level cases, including salt combinations, solvent sets, concentration ranges, the violated gate, and any nearby counterexample that passed,

so that proposed corrections do not overgeneralize from a single small batch. Each iteration also logs six lightweight metrics covering pass efficiency, progress toward the target count, weakest-gate margin, gate-level attainment, compositional diversity, and mixed-salt success rate. These metrics are treated as process signals rather than proof of causal improvement, since a higher pass rate may simply reflect an easier part of the search space. Any change proposed by the reflector to retrieval priorities, search priors, or rejection rules must pass a guard that checks the failure is real, the target is approved, and the task specification is unchanged. At most two such changes are accepted per iteration, and a change is rolled back automatically if the next iteration produces both fewer newly valid candidates and no improvement on the failure it was meant to address.

Finally, we illustrate each agent's responsibilities in detail. SAGE organizes the workflow into five agents that communicate through schema-validated artifacts and fixed tool interfaces instead of unconstrained conversation. The *planner* initializes each task-specific iteration from the immutable specification $\mathcal{S}_t$, gate set $\mathcal{G}_t$, remaining budget $\mathcal{B}_t$, and asynchronous scheduler state. This fixes the evaluation allocation and prevents intermediate results from changing the objective or acceptance rules. The *retriever* turns the previous archive and actionable failure set $F_{t,i-1}$ into prospective or failure-specific queries, collecting source-linked evidence on relevant mechanisms, experimental conditions, applicability, and counterexamples while reserving context for the gates that actually failed. The *proposer* combines this evidence with the current task-private archive and design policy $\pi_{t,i}$ to produce bounded and testable qualitative hypotheses. These hypotheses identify target gates, component interactions, plausible mechanisms, and search operators, but never invent concentrations, predicted properties, thresholds, or PASS labels. The *evaluator* then applies deterministic composition, duplicate, legality, and constraint checks together with bounded risk and quality-control rules before exporting admissible candidates to the prescribed MD or DFT protocols. As a result, numerical values and physical decisions remain outside the agents' authority. The *reflector* interprets the returned physical outcomes and failure attributions, preserves successful counterexamples, and compiles only guarded, failure-specific changes $\Delta\pi_{t,i}$ to agent skills, task prompts, retrieval priorities, and tool-use strategies. It also applies rollback protection when a proposed update does not address its targeted failure. At the end of each iteration, externally measured results are stored in the immutable task-specific archive, while an accepted update produces $\pi_{t,i+1} = \mathrm{Commit}(\pi_{t,i}, \Delta\pi_{t,i})$ for the next retrieval, hypothesis, and candidate-search loop. Throughout this process, the task specification, gate thresholds, and past measurements remain unchanged.

### A.4 Detailed Experimental Setup

We evaluate SAGE on five industrial electrolyte formulation tasks that represent common battery design scenarios: general requirements, fast charge, low temperature, high temperature, and high voltage design (task 1 to task 5). Each task is defined by a fixed task specification $\mathcal{S}_t = (\Omega_t, \mathcal{O}_t, \mathcal{G}_t, \mathcal{B}_t)$, which determines the admissible salts, solvents, and additives, the target objectives, the feasibility gates $\mathcal{G}_t$, and the physical evaluation budget. All numerical gates follow the shared bootstrap confidence rule in Equation (4). For task 5, the reference formulation identity and stoichiometry are part of the locked control artifact and are not changed per candidate.

Specifically, task 1 permits registered, compatibility-checked formulations containing at least one electrolyte salt and requires $\sigma_{\mathrm{NE}}^{\mathrm{screen}} \geq 2.0$ mS/cm-equivalent, density in $[0.80, 1.45]$ g/cm$^3$, coordination number in $[3.0, 9.0]$, largest persistent ion-cluster fraction no greater than $0.75$, and apparent viscosity no greater than $10.0$ mPa·s. These criteria capture the interplay among ion transport, solvation, and aggregation in bulk electrolytes (Giffin, 2022; Mistry et al., 2022). Task 2 restricts total salt concentration to $[0.8, 2.5]$ mol/L and formulation size to 2–7 components, raises the transport threshold to $\sigma_{\mathrm{NE}}^{\mathrm{screen}} \geq 8.0$ mS/cm-equivalent, and requires lithium ions to contribute at least $0.40$ of the uncorrelated NE transport estimate. It also requires density in $[1.00, 1.55]$ g/cm$^3$, coordination number in $[3.5, 6.0]$, largest persistent ion-cluster fraction no greater than $0.80$, and apparent viscosity no greater than $12.0$ mPa·s, reflecting fast-charge-oriented bulk screening (Logan & Dahn, 2020; Song et al., 2025). Task 3 considers formulations with one or more lithium salts and a total salt concentration in $[0.5, 2.0]$ mol/L and no more than seven components. Candidates are cooled through

298.15, 273.15, and 253.15 K and evaluated at 253.15 K ($-20°$C). The task requires $\sigma_{\mathrm{NE}}^{\mathrm{screen}} \geq 0.50$ mS/cm-equivalent, density in $[0.85, 1.45]$ g/cm$^3$, coordination number in $[3.5, 8.0]$, largest persistent ion-cluster fraction no greater than $0.50$, and apparent viscosity no greater than $50.0$ mPa·s. The relaxed viscosity threshold reflects low-temperature flow resistance, while the stricter cluster constraint excludes strongly aggregated or insufficiently relaxed systems (Liu & Wang, 2023; Liu et al., 2025; Peng et al., 2025). Task 4 uses one or more lithium salts and a total salt concentration in $[0.8, 2.0]$ mol/L, allows at most seven components, and evaluates each candidate at both 298.15 and 328.15 K. At 328.15 K (55°C), the candidate must satisfy $\sigma_{\mathrm{NE}}^{\mathrm{screen}} \geq 5.0$ mS/cm-equivalent and coordination number in $[3.5, 8.0]$, while the relative change in coordination number between the two temperatures must not exceed $0.15$. It also requires density in $[0.80, 1.50]$ g/cm$^3$, largest persistent ion-cluster fraction no greater than $0.65$, and apparent viscosity no greater than $8.41$ mPa·s, jointly assessing high-temperature transport and structural stability across temperatures (Liu & Wang, 2023; Wan et al., 2024). Task 5 considers formulations with one or more lithium salts and a total salt concentration in $[0.8, 2.0]$ mol/L, together with 2–5 solvents, no more than three additives, and no more than seven total components. A candidate must satisfy $\sigma_{\mathrm{NE}}^{\mathrm{screen}} \geq 3.0$ mS/cm-equivalent, density in $[0.90, 1.65]$ g/cm$^3$, coordination number in $[3.5, 10.0]$, largest persistent ion-cluster fraction no greater than $0.80$, and apparent viscosity no greater than $15.0$ mPa·s. It must also satisfy the `T5_DFT_VIP` criterion, requiring the candidate cluster VIP to be no more than $0.30$ eV below a matched-protocol reference. The locked task 5 reference is derived from the designated control formulation. For both candidate and reference, three non-adjacent local anion–solvent or anion–additive cluster snapshots are extracted from MD trajectories, periodic images are removed, and the same $\Delta$SCF protocol is applied using M06-HF/def2-SVPD without ionized-geometry relaxation. This criterion serves as a relative electronic-stability screen for high-voltage-oriented formulations rather than a direct measure of the electrochemical window (Ren et al., 2019a; Xiao et al., 2024). All gates are applied together with shared simulation quality-control requirements. Passing them does not establish coulombic efficiency, cycle life, interfacial stability, or a realized electrochemical window. Rather, it shows that a frozen formulation satisfies the task-specific transport and liquid-state structural screens and, for task 5, the relative cluster VIP screen under a fixed computational protocol.

For comparisons conducted on the same tasks and under matched candidate budgets, we keep the task specification $\mathcal{S}_t$, the gate set $\mathcal{G}_t$, the physical evaluation protocol, and the decision rule in Equation (4) fixed across all methods, including SAGE. This ensures that any difference in PASS rate reflects search quality rather than looser thresholds or a larger search budget. Each task runs for 16 iterations, with five candidates physically evaluated per iteration, yielding a total of 80 physically evaluated candidates per task. To support retrieval, candidate construction, and novelty auditing, we use several public databases: Bamboo-Mixer (Yang et al., 2026), CALiSol-23 (de Blasio et al., 2024), the SCAN candidate atlas (Wang & You, 2026), MolSets (Zhang et al., 2023), and ElectrolyteGPT (Kim et al., 2026). These resources are used for retrieval, candidate-space construction, and candidate-level novelty auditing. They are never allowed to assign pass or fail labels. Admission to $\mathcal{V}_{t,i}$ in Equation (2) depends only on the fixed physical evaluation protocol and the bootstrap rule in Equation (4). This keeps acceptance criteria consistent and novelty comparisons fair across all methods. For baselines, we fix the task specification, gate set, and budget, while allowing backbone model and agentic structure to vary by method. We evaluate multiple backbones: GPT-5.6 Terra (Singh et al., 2025), DeepSeek V4 Flash (Xu et al., 2026), Qwen3.8-27B (Yang et al., 2025), LLaMat-3 (Mishra et al., 2024), and NatureLM 8-7B (Xia et al., 2025). These models cover both general-purpose and scientific language models, allowing us to examine whether domain-specific pretraining improves electrolyte discovery. All adapted baseline agents were given access to the same deterministic programs and tool interfaces as SAGE. They span different model families and deployment settings, reducing the risk that the comparison reflects the behavior of a single provider or architecture. Each backbone is tested under two settings: LLM with web search, where the model receives one retrieved snippet before directly proposing candidates, and agent with tool use, where the same model operates a bounded loop over web search, candidate search, and deterministic constraint checking. This paired design separates gains from model knowledge and gains from agentic structure. We also adapt three state-of-the-art scientific agentic workflows: ChemCrow (M. Bran et al., 2024), ChemGraph (Pham et al., 2026), and MatSciAgent (Chaudhari et al., 2026). They were selected because they represent complementary control styles, including tool-oriented reasoning, planner–worker coordination, and specialist routing. We preserve their original control styles while instantiating all three with the same GPT-5.6 backbone (Singh et al., 2025) and the same audited tool and gate interface as SAGE. This makes the comparison consistent in protocol and resource budget.

The ablation study consists of two parts: base model swapping and component ablations. The backbone model swap study in Figure 5 reruns SAGE with multiple alternative LLM backbones of different sizes while keeping the workflow and evaluation protocol unchanged. The component removal study in Figure 7 evaluates `strict_stateless_no_update` (self-improvement), `text_reflection_only` (text reflection), `no_domain_routing` (failure attribution), `feedback_blind` (MD feedback), `no_skill_consumption` (agent skill), `no_interaction_memory` (MD archive), `no_guard_consumption` (guard check), `no_tool_use_consumption` (tool-use strategy), and `no_rollback` (rollback mechanism). Each variant removes or restricts one mechanism to assess the contributions of self-improvement, MD feedback, structured reflection, archived memory, and guarded policy updates. All ablated variants use the same task specifications, candidate budgets, component catalog, physical constraints, MD protocol, and evaluation criteria, with histories and feedback isolated across runs. The knowledge transfer analysis in Figure 6 compares fresh runs initialized with structured knowledge from original tasks against an unseeded iteration 2 baseline and the fully self-improved iteration 16 result. This setup separates the effect of transferred knowledge from continued in-task self-improvement.

Moreover, SAGE registers multiple tool interfaces, including `knowledge_retriever`, `constraint_checker`, `nsga2_engine`, `external_validator`, and `novelty_checker`. The `knowledge_retriever` searches registered literature and data sources using task objectives, chemical species, and prior failure signatures to provide traceable evidence. The `constraint_checker` deterministically verifies component identity, normalization, concentration, compatibility, and structural validity, with only explicitly permitted repairs. The `nsga2_engine` converts qualitative agent hypotheses into valid numerical formulations and performs constrained multi-objective search, including Pareto selection, mutation, recombination, and deduplication. The `novelty_checker` audits each candidate at the formulation level against a local formulation index, task-specific submission history, and the reference formulation databases to detect duplicates and highly similar compositions within the indexed search space. After deterministic constraint and novelty checks, the evaluator selects the top five candidates under predefined programs and submits them through the `external_validator`, which provides a typed interface to isolated MD and DFT calculations. These quality-controlled results are the sole authority for physical pass/fail decisions. Thus, the LLM proposes mechanistic hypotheses and search directions, while the tools handle retrieval, numerical generation, validation, novelty auditing, and physical-computation handoff, ensuring reproducibility, traceability, and clear separation of scientific authority. Below, we present an illustrative prompt template used in our tasks.

**Prompt Template Example**

**TEMPLATE**

```
task:
  task_id: <T01--T05>
  title: <human-readable task title>
  objective: <scientific design objective>

  optimization_objectives:
    - name: <property name>
      direction: <maximize|minimize|target>
      unit: <unit or null>

  allowed_pool:
    roles: [salt, solvent, additive]
    component_ids: <registered, compute-ready IDs only>
  composition_basis: <A|B|task-specific>

  constraints:
    pre_filter:
      - id: <constraint ID>
        verifier: <identity|composition|compatibility|structure>
```

```
        rule: <deterministic rule>
    md_dft:
      - id: <gate ID>
        metric: <measured MD or DFT metric>
        operator: <min|max|eq|between_inclusive>
        threshold: <frozen value or interval>
        unit: <unit>

  soft_priors:
    - <qualitative scientific prior>
  claim_boundary:
    - <conclusions not supported by the validation protocol>

campaign:
  iteration_index: <integer beginning at 1>
  working_pool_size: 80
  authoritative_candidate_pool_size: 20
  external_validation_top_k: 5
  total_rounds: <program-fixed integer>
  quantity_gap: <program-computed integer>
  remaining_candidate_budget: <program-computed integer>
  remaining_external_budget: <program-computed integer>
  regeneration_strategy: <null|archive_guided|quota_sampling>
  retrieval_queries: <task-level, gate-specific, and bounded model-
    augmented queries>

initial_context:
  failure_signatures: <task-specific failures or empty list>
  real_task_local_archive: <quality-controlled results or empty list>
  learned_skills: <guard-approved skills or empty list>
  evidence_cards: <retriever-provided evidence or empty list>
  allowed_component_ids: <registered, compute-ready IDs only>
  excluded_composition_hashes: <previously evaluated hashes>

guarded_policy:
  numeric_budget_authority: program_only
  chemistry_value_authority: typed_composer_only
  pass_fail_authority: measured_MD_DFT_and_QC_only

  allowed_search_operators:
    - seed_from_evidence
    - seed_from_low_discrepancy_design
    - mutate_ratio
    - swap_component
    - split_primary_salt
    - merge_or_drop_component
    - recombine_compatible_parents
    - probe_failed_gate_boundary

  prohibited_model_actions:
    - change the TaskSpec, budget, gates, or thresholds
    - invent components, structures, evidence, or results
    - output exact fractions or concentrations as hypotheses
    - assign PASS or FAIL before MD/DFT and QC
    - modify or overwrite historical physical results
    - claim properties beyond the stated claim boundary

required_action:
  - follow the role-specific system contract
  - use only the supplied task context and registered components
  - preserve evidence and parent-artifact references
  - return only the required role-specific JSON object
```

Table 3: Ten representative formulations designed by SAGE for task 1.

| ID | Salt Content (mol%) | Solvents (mol%) | Additive (mol%) |
|---|---|---|---|
| 1-01 | NaFSI 4.39; NaTFSI 1.56 | EMC 1.03; DME 25.35; EC 45.11; MBE 21.77 | DTD 0.80 |
| 1-02 | NaFSI 3.31; NaTFSI 2.23 | EMC 1.03; DME 25.46; EC 45.30; MBE 21.87 | DTD 0.80 |
| 1-03 | NaFSI 4.28; NaTFSI 2.20 | EMC 1.02; DME 25.20; EC 44.85; MBE 21.65 | DTD 0.80 |
| 1-04 | NaFSI 3.73; NaTFSI 2.22 | EMC 1.03; DME 25.35; EC 45.11; MBE 21.77 | DTD 0.80 |
| 1-05 | NaFSI 3.78; NaTFSI 2.17 | EMC 1.03; DME 25.35; EC 45.11; MBE 21.77 | DTD 0.80 |
| 1-06 | NaFSI 3.51; LiTFSI 2.61 | Diglyme 54.2; DEC 3.62; DPE 35.38 | PS 0.68 |
| 1-07 | LiFSI 9.37; $LiBF_4$ 3.42 | MPE 84.9 | TMSP 2.31 |
| 1-08 | LiFSI 8.31; $LiBF_4$ 2.61 | DIOX 87.06 | DTD 2.02 |
| 1-09 | $LiPF_6$ 8.31; $LiBF_4$ 2.61 | MPE 87.06 | DTD 2.02 |
| 1-10 | LiFSI 9.48; $LiBF_4$ 2.98 | MPE 85.87 | PS 1.68 |

Table 4: Ten representative formulations designed by SAGE for task 2.

| ID | Salt Concentration (mol/L) | Solvent Molar Ratio | Additive (mol) |
|---|---|---|---|
| 2-01 | LiTFSI 0.68; LiFSI 0.37 | MA 15.9; PC 16.3; DEE 55.2; GBL 12.7 | TMSP 3.00 |
| 2-02 | LiFSI 0.62; $LiBF_4$ 0.26 | DOL 8.8; AN 14.9; BTFE 54.0; PC 22.3 | TMSP 3.00 |
| 2-03 | LiTFSI 1.64; $LiBF_4$ 0.71 | BTFE 14.0; AN 52.1; METHF 33.9 | VC 3.00 |
| 2-04 | LiFSI 1.64; $LiBF_4$ 0.71 | BTFE 13.5; AN 53.9; METHF 32.6 | VC 3.00 |
| 2-05 | LiTFSI 1.64; $LiBF_4$ 0.71 | MPE 13.5; AN 53.9; METHF 32.6 | VC 3.00 |
| 2-06 | LiTFSI 1.64; $LiBF_4$ 0.71 | BTFE 13.4; AN 53.9; METHF 32.7 | VC 3.32 |
| 2-07 | LiTFSI 1.64; $LiBF_4$ 0.71 | BTFE 12.9; AN 51.7; METHF 35.3 | VC 3.32 |
| 2-08 | LiFSI 1.66; LiTFSI 0.67 | DEE 53.9; PC 46.1 | VC 3.00 |
| 2-09 | $LiBF_4$ 1.06; LiTFSI 0.65 | MTFA 31.0; DPE 44.6; DMTHF 24.4 | DTD 3.00 |
| 2-10 | LiFSI 0.80; $LiBF_4$ 0.57 | DMC 23.9; PC 33.6; BTFE 19.3; FEC 23.1 | DTD 2.46 |

### A.5 Representative Formulation Examples

This section provides concrete examples of designed formulations for the five task settings reported in the paper. Composition units follow the composer's source representation rather than a single unified basis. In task 1, salts, base solvents, and additives are all reported as mol% of the full mixture and therefore share the same denominator. In tasks 2–5, each salt is reported by its nominal concentration (mol $L^{-1}$) with respect to the final mixture volume specified by the composer. Base-solvent proportions are normalized to sum to 100 within the base-solvent pool, and additive loading is reported as mol per 100 mol of base solvents. Therefore, these quantities should not be summed or directly interconverted without the explicit volume definition and integerization procedure used to construct the simulation box. A glossary for the tables is provided in Table 8.

**Task 1: General Requirements.** As shown in Table 3, task 1 is intentionally general and is designed to test whether the workflow can preserve valid component identities, composition bases, compatibility, provenance, and physical screening across chemically distinct formulation families. The selected candidates cover sodium-salt mixtures, sodium–lithium mixtures, and lithium-salt mixtures, requiring the same representation and evaluation pipeline to handle different charge carriers, anion chemistries, solvent coordination environments, and additive roles without implicitly assuming a lithium-only system. The first five rows define a controlled NaFSI/NaTFSI family in which the solvent environment is nearly fixed while the salt ratio varies, making them useful for examining how anion identity and relative salt balance influence ion association without the confounding effect of a different solvent system. The sixth row shifts both the carrier-ion combination and the solvent regime, serving as a cross-family general requirement. Rows 1-07 to 1-10 illustrate another design pattern,

Table 5: Ten representative formulations designed by SAGE for task 3.

| ID | Salt Concentration (mol/L) | Solvent Molar Ratio | Additive (mol) |
|---|---|---|---|
| 3-01 | LiFSI 1.06; LiTFSI 0.73 | MA 21.7; FEC 9.4; DME 29.9; MTBE 39.0 | — |
| 3-02 | $LiPF_6$ 0.83; $LiBF_4$ 0.35 | DME 18.2; MBE 74.0; MA 7.8 | — |
| 3-03 | $LiPF_6$ 0.83; LiTFSI 0.35 | DME 18.5; DMTHF 73.7; MA 7.9 | — |
| 3-04 | LiTFSI 0.83; LiFSI 0.35 | DME 19.4; MBE 73.6; MA 7.0 | — |
| 3-05 | $LiPF_6$ 0.83; LiTFSI 0.35 | MTFA 19.2; MBE 72.7; MA 8.2 | — |
| 3-06 | $LiPF_6$ 0.83; LiFSI 0.35 | DEE 19.3; MBE 73.3; MA 7.4 | — |
| 3-07 | $LiPF_6$ 0.83; LiTFSI 0.29 | DME 18.2; MBE 74.0; EMC 7.8 | — |
| 3-08 | $LiPF_6$ 0.83; LiFSI 0.35 | DME 19.2; MBE 72.7; DIOX 8.2 | — |
| 3-09 | LiFSI 1.04 | DME 67.9; EA 24.9; MA 7.2 | PS 3.00 |
| 3-10 | LiFSI 1.04 | DME 67.1; EA 24.6; MA 8.3 | PS 3.00 |

Table 6: Ten representative formulations designed by SAGE for task 4.

| ID | Salt Concentration (mol/L) | Solvent Molar Ratio | Additive (mol) |
|---|---|---|---|
| 4-01 | $LiBF_4$ 1.11; $LiPF_6$ 0.37 | Triglyme 27.5; SL 30.7; METHF 41.0; GBL 0.9 | PS 3.00 |
| 4-02 | $LiBF_4$ 1.03; LiTFSI 0.45 | THF 27.5; SL 30.7; METHF 41.0; GBL 0.8 | PS 3.00 |
| 4-03 | $LiBF_4$ 0.90; LiFSI 0.58 | Triglyme 27.5; SL 30.7; METHF 41.0; GBL 0.9 | PS 3.00 |
| 4-04 | $LiBF_4$ 1.04; LiTFSI 0.44 | Triglyme 27.5; SL 30.7; METHF 41.0; GBL 0.9 | PS 3.00 |
| 4-05 | $LiBF_4$ 0.93; $LiPF_6$ 0.56 | Triglyme 27.5; SL 30.7; METHF 41.0; GBL 0.9 | PS 3.00 |
| 4-06 | $LiBF_4$ 1.15; LiTFSI 0.34 | Triglyme 27.5; SL 30.7; METHF 41.0; GBL 0.9 | PS 3.00 |
| 4-07 | $LiBF_4$ 0.97; LiFSI 0.51 | Triglyme 27.5; SL 30.7; METHF 41.0; GBL 0.9 | PS 3.00 |
| 4-08 | $LiBF_4$ 0.96; $LiPF_6$ 0.52 | Triglyme 27.5; SL 30.7; METHF 41.0; GBL 0.8 | PS 3.00 |
| 4-09 | LiTFSI 1.14; $LiBF_4$ 0.31 | EMC 8.8; DOL 8.9; EA 82.3 | TMSP 3.00 |
| 4-10 | $LiBF_4$ 0.99; LiTFSI 0.49 | Triglyme 27.5; SL 30.7; METHF 41.0; GBL 0.9 | PS 3.00 |

where a minor secondary lithium salt is combined with a dominant ether solvent and a low-level phosphorus- or sulfur-containing additive. From a physical-chemistry perspective, these formulations reflect a central multi-objective tradeoff: stronger coordination or higher solvation can improve dissolution and stabilize local ion environments, but excessive association may reduce the fraction of independently mobile charge carriers (Mistry et al., 2022). By contrast, weakly coordinating ether-rich mixtures may enhance mobility, but can also reshape anion participation, viscosity, and cluster formation, meaning that weak solvation cannot be captured by a single solvent descriptor (Wang et al., 2023). SAGE does not aim to identify a universally superior family. Instead, its contribution is to explore across these families while preserving the component roles, compositional basis, and physical evidence needed to interpret both successes and failures.

**Task 2: Fast Charge.** As presented in Table 4, task 2 targets high bulk transport together with a minimum lithium contribution to the non-correlated transport estimate. All examples contain two lithium salts, while the formulation family shifts from four-solvent mixtures to AN/METHF-rich ternary systems and finally to a compact DEE/PC mixture. The recurring LiTFSI/$LiBF_4$ and LiFSI/$LiBF_4$ combinations are informative rather than redundant, as they probe how anion chemistry, solvent donor strength, and additive level jointly shape the balance between overall mobility and lithium-ion transport. The AN/METHF/BTFE family is especially useful because it combines a polar nitrile environment with a fluorinated ether and a second solvent of different donor character, creating conditions that can redistribute lithium solvation and alter cation-anion mobility. In this setting, stronger coordination may improve dissociation in one regime while also increasing solvation drag on $Li^+$, which is why maximizing total transport alone is insufficient for this task. Additives such as TMSP or VC are therefore kept as separate chemical roles rather than absorbed into the solvent ratio,

Table 7: Ten representative formulations designed by SAGE for task 5.

| ID | Salt Concentration (mol/L) | Solvent Molar Ratio | Additive (mol) |
|---|---|---|---|
| 5-01 | $LiBF_4$ 1.04; $LiPF_6$ 0.69 | GBL 32.0; METHF 28.3; DEE 16.2; DMC 23.4 | DTD 3.00 |
| 5-02 | $LiPF_6$ 1.11; $LiBF_4$ 0.44 | SL 1.1; FEC 27.9; DOL 46.9; AN 24.0 | PS 3.00 |
| 5-03 | $LiPF_6$ 1.11; $LiBF_4$ 0.44 | THF 1.1; FEC 27.2; METHF 48.4; AN 23.3 | VC 3.00 |
| 5-04 | $LiPF_6$ 1.11; $LiBF_4$ 0.44 | SL 1.1; FEC 27.2; METHF 48.4; AN 23.3 | DTD 3.00 |
| 5-05 | $LiPF_6$ 1.11; LiFSI 0.44 | SL 1.1; FEC 27.2; METHF 48.4; AN 23.3 | PS 3.00 |
| 5-06 | $LiPF_6$ 1.11; $LiBF_4$ 0.44 | SL 1.1; FEC 27.9; METHF 46.9; AN 24.0 | PS 3.00 |
| 5-07 | $LiPF_6$ 1.11; $LiBF_4$ 0.44 | SL 1.2; FEC 28.6; METHF 45.7; AN 24.6 | VC 3.00 |
| 5-08 | $LiPF_6$ 1.11; $LiBF_4$ 0.44 | EC 1.1; FEC 27.2; METHF 48.4; AN 23.3 | DTD 3.00 |
| 5-09 | $LiPF_6$ 1.29; $LiBF_4$ 0.44 | SL 1.1; FEC 27.2; METHF 48.4; AN 23.3 | PS 3.00 |
| 5-10 | $LiPF_6$ 1.11; $LiBF_4$ 0.44 | SL 1.1; FEC 27.2; METHF 48.4; AN 23.3 | PS 3.00 |

preserving the distinction between bulk-transport design and interphase-oriented additive design (Park et al., 2023). Rows 2-03 to 2-09 form a set of closely related local variants, which is valuable for mechanism-oriented analysis because small substitutions or ratio changes can be interpreted against a shared physical background.

**Task 3: Low Temperature.** As presented in Table 5, task 3 examines the formulation problem at $-20\,^{\circ}\mathrm{C}$. The ten examples exhibit two main compositional patterns. The first eight formulations use mixed lithium salts with diverse ether/ester-based co-solvent mixtures and contain no additives. In contrast, the last two are closely related single-salt variants based on LiFSI, DME, EA, and MA, both with PS as the only additive. Across the first group, DME and MBE appear frequently, while MA is also present in most formulations; other solvents occur in individual recipes. These variations indicate that SAGE explores multiple combinations of salt identity, salt loading, and solvent composition rather than converging on a single recipe family. The two PS-containing variants represent a more localized compositional motif and should not be interpreted as evidence that additive stabilization is a general feature of the task 3 formulations. Overall, the pass criterion should be interpreted as a physical-simulation-based screen of transport, coordination structure, compatibility, and related behavior under the specified protocol, rather than as direct evidence of experimentally measured low-temperature performance.

**Task 4: High Temperature.** As presented in Table 6, task 4 evaluates transport at $55\,^{\circ}\mathrm{C}$ while constraining the relative change in local coordination between the low- and high-temperature endpoints. Most selected formulations keep the solvent environment nearly fixed and vary only the salt pair and salt ratio, making this family well suited for studying the thermal robustness of local solvation structure. Under this design, changes in the response can be attributed primarily to salt identity and composition while the dominant solvent background remains stable. Compared with task 3, the central challenge here is different. Higher temperature generally enhances molecular motion and can improve transport, but it can also shift coordination equilibria, accelerate ligand exchange, and increase the sensitivity of local solvation structure to anion chemistry. The relative CN-change criterion therefore tests whether a formulation maintains a similar coordination regime after heating, while the high-temperature CN window and viscosity constraint ensure that the endpoint remains physically reasonable. The repeated Triglyme/SL/METHF background is especially useful for comparing salt-ratio effects, whereas the EA-rich row provides a deliberate contrast at the formulation-family level. These calculations are relevant to the broader electrolyte design problem, where transport and interphase behavior depend jointly on solvent environment and salt chemistry (Zhang et al., 2024; Wan et al., 2024; Song et al., 2025).

**Task 5: High Voltage.** As presented in Table 7, task 5 combines bulk transport with a relative cluster VIP screen. Its selected formulations are compositionally complex by design, commonly containing two lithium salts, four solvents, and one additive. Most rows form a $LiPF_6$/$LiBF_4$ base with FEC, AN, METHF, and a small SL fraction, while the family explores solvent replacement, salt replacement, $LiPF_6$ loading, and additive identity. The design is therefore not a simple search for a single high-ionization solvent. It searches for local combinations in which transport, solvation, aggregation,

viscosity, and the electronic stability proxy remain jointly admissible. Several physical-chemistry insights follow from this family. FEC is a major solvent fraction in these records, not a trace additive, so its effect must be interpreted together with AN, METHF, and the salt pair. AN can contribute strong dipolar solvation and transport-relevant fluidity, while fluorinated and carbonate components can change local anion–solvent interactions and interphase-relevant chemistry. $LiBF_4$, $LiPF_6$, and LiFSI also differ in anion size, charge delocalization, coordination propensity, and susceptibility to ion pairing, so changing the second salt can alter both the dominant cluster selected from MD and the bulk transport response. The small SL or EC replacement is consequently a chemically meaningful perturbation even when its nominal ratio is near one percent, because it can change the identity and local environment of the cluster used for the VIP calculation.

Essentially, these five formulation tasks illustrate why the search problem cannot be reduced to a single scalar property. General requirements focus on chemical validity across heterogeneous component roles, fast charge balances total and lithium-specific transport, low temperature trades mobility against aggregation and viscosity, high temperature constrains both transport and structural stability across temperature, and high voltage couples bulk transport with a local electronic-stability proxy. The resulting PASS compositions are thus best viewed as structured hypotheses for feasible regions in a coupled formulation space. Their novelty lies in the specific combinations of salts, solvents, ratios, and additives that jointly satisfy the computational criteria, rather than in any claim that each component is individually new or universally optimal. This interpretation is also consistent with the high prevalence of mixed-salt formulations among PASS candidates across several tasks. Mixed salts do not by themselves prove synergy, but they do enlarge the accessible tradeoff space among dissociation, solvation structure, viscosity, and aggregation, while also introducing possible penalties such as stronger ion association or interfacial incompatibility. The value of SAGE is that it searches these competing effects within a single auditable pipeline, using the same hard constraints for both database-derived and newly proposed formulations.

Table 8: Glossary of component abbreviations and full names. Roles (salt, solvent, additive) are indicated by the group headers below.

| **Abbreviation** | **Full Name** |
|---|---|
| | *Salts* |
| $LiBF_4$ | lithium tetrafluoroborate |
| LiFSI | lithium bis(fluorosulfonyl)imide |
| $LiPF_6$ | lithium hexafluorophosphate |
| LiTFSI | lithium bis(trifluoromethanesulfonyl)imide |
| NaFSI | sodium bis(fluorosulfonyl)imide |
| NaTFSI | sodium bis(trifluoromethanesulfonyl)imide |
| | *Solvents* |
| AN | acetonitrile |
| BTFE | bis(2,2,2-trifluoroethyl) ether |
| DEC | diethyl carbonate |
| DEE | diethyl ether |
| DIOX | 1,4-dioxane |
| DMC | dimethyl carbonate |
| DME | 1,2-dimethoxyethane |
| DMTHF | 2,5-dimethyltetrahydrofuran |

Table 8 (continued)

| Abbreviation | Full Name |
|---|---|
| DOL | 1,3-dioxolane |
| DPE | dipropyl ether |
| Diglyme | diethylene glycol dimethyl ether |
| EA | ethyl acetate |
| EC | ethylene carbonate |
| EMC | ethyl methyl carbonate |
| FEC | fluoroethylene carbonate |
| GBL | gamma-butyrolactone |
| MA | methyl acetate |
| MBE | methyl butyl ether |
| METHF | 2-methyltetrahydrofuran |
| MPE | methyl propyl ether |
| MTBE | methyl tert-butyl ether |
| MTFA | methyl trifluoroacetate |
| PC | propylene carbonate |
| SL | sulfolane |
| THF | tetrahydrofuran |
| Triglyme | triethylene glycol dimethyl ether |
| | *Additives* |
| DTD | dioxathiolane dioxide |
| PS | 1,3-propane sultone |
| TMSP | tris(trimethylsilyl) phosphite |
| VC | vinylene carbonate |

Table 9: Novelty results for electrolyte formulation design. Values denote the percentage of candidates identified as chemically novel relative to the reference corpus under a uniform evaluation protocol across all methods.

| **Method** | | **Task 1** | **Task 2** | **Task 3** | **Task 4** | **Task 5** | **Avg. Novelty** |
|---|---|---|---|---|---|---|---|
| *Backbone* | *Agentic Design* | *General* | *Fast Charge* | *Low Temp.* | *High Temp.* | *High Volt.* | |
| | | | *Large Language Models* | | | | |
| GPT-5.6 Terra | LLM + web search | 28.75 | 23.75 | 11.25 | 18.75 | 5.00 | 17.50 |
| LLaMat-3 | LLM + web search | 13.75 | 6.25 | 0.00 | 0.00 | 1.25 | 4.25 |
| NatureLM 8-7B | LLM + web search | 15.00 | 16.25 | 5.00 | 10.00 | 0.00 | 9.25 |
| DeepSeek V4 Flash | LLM + web search | 17.50 | 17.50 | 11.25 | 17.50 | 0.00 | 12.75 |
| Qwen3.8-27B | LLM + web search | 20.00 | 5.00 | 3.75 | 0.00 | 1.25 | 6.00 |
| | | | *General-Purpose Agentic Workflow* | | | | |
| GPT-5.6 Terra | Agent with tool use | 32.50 | 16.25 | 13.75 | 21.25 | 3.75 | 17.50 |
| LLaMat-3 | Agent with tool use | 17.50 | 3.75 | 1.25 | 2.50 | 2.50 | 5.50 |
| NatureLM 8-7B | Agent with tool use | 21.25 | 15.00 | 6.25 | 8.75 | 0.00 | 10.25 |
| DeepSeek V4 Flash | Agent with tool use | 30.00 | 22.50 | 12.50 | 20.00 | 3.75 | 17.75 |
| Qwen3.8-27B | Agent with tool use | 12.50 | 16.25 | 3.75 | 2.50 | 0.00 | 7.00 |
| | | | *Adapted Agentic Workflows* | | | | |
| ChemCrow | ReAct Controller | 32.50 | 17.50 | 8.75 | 15.00 | 0.00 | 14.75 |
| ChemGraph | Planner Worker | 23.75 | 22.50 | 10.00 | 16.25 | 3.75 | 15.25 |
| MatSciAgent | Router Specialist | 25.00 | 8.75 | 7.50 | 11.25 | 3.75 | 11.25 |
| DeepSeek V4 Flash | **SAGE (Ours)** | **70.00** | **77.50** | **55.00** | **52.50** | **76.25** | **66.25** |

### A.6 Evaluation of Formulation Novelty

To further evaluate SAGE, we adopt novelty as an additional metric that measures whether designed formulations among the PASS candidates are missing from a fixed reference corpus. However, the lack of an exact compositional match does not, by itself, imply a novel chemical family or formulation region. Empirically, the novelty results in Table 9 and the main results in Table 2 show that SAGE's advantage is not limited to a single evaluation axis, but remains consistent across both validity and originality. In other words, SAGE achieves stronger overall performance while also producing substantially more chemically novel candidates. This makes a simple explanation unlikely, namely that the method succeeds mainly by reusing known and already validated formulations, which would typically reduce novelty. Instead, the results indicate that SAGE is able to explore new regions of formulation space while still maintaining strong performance under the fixed evaluation protocol.

One reason is that SAGE does not rely on a single-shot proposal process, but iteratively revises its design policy in response to physically grounded failure signals, which helps it move beyond familiar high-frequency formulation patterns. In addition, its structured decomposition of salts, solvents, additives, and competing constraints makes it easier to identify compensatory multicomponent edits that are less likely to appear in standard database priors. This is especially critical in coupled design spaces, where improving one property often requires coordinated adjustments in other components rather than isolated substitutions. As a result, SAGE is better positioned to reach sparse but physically plausible regions of formulation space that simpler retrieval-based or one-pass methods may never explore. Table 9 further shows that SAGE achieves the highest novelty on every task, with an average novelty of $66.25\%$, far exceeding all compared LLM, general-purpose agentic, and adapted scientific-agent baselines. The gap is especially pronounced in constrained settings such as task 5, where most baselines produce almost no novel candidates while SAGE reaches $76.25\%$, indicating a much stronger ability to move beyond high-frequency known recipes under strict physical requirements. Overall, these results suggest that SAGE's gains come not from memorization or familiar recipe recombination, but from structured feedback and iterative policy refinement that support physics-informed exploration of new composition regions.

### A.7 Additional Main Results and Ablations

**Self-Improvement Traces.** Figures 8 to 12 visualize the iterative search traces for the five tasks over 16 iterations. Each trace records how candidates evolve across iterations as constraints are progressively satisfied. Across settings, the trajectories show gradual movement toward feasible regions, although local fluctuations remain common during exploration. This pattern suggests that physical feedback helps correct the subsequent search strategy toward more promising formulations without changing the underlying task criteria.

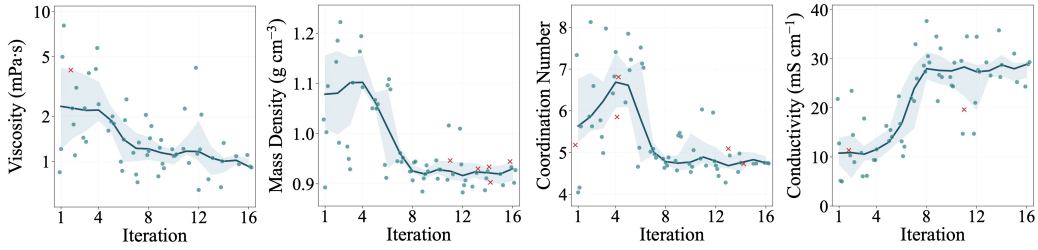



Figure 8: Self-improvement trace for the general requirements setting.

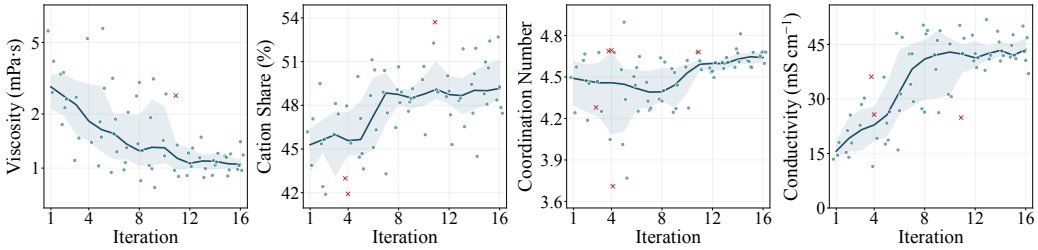



Figure 9: Self-improvement trace for the fast-charge setting.

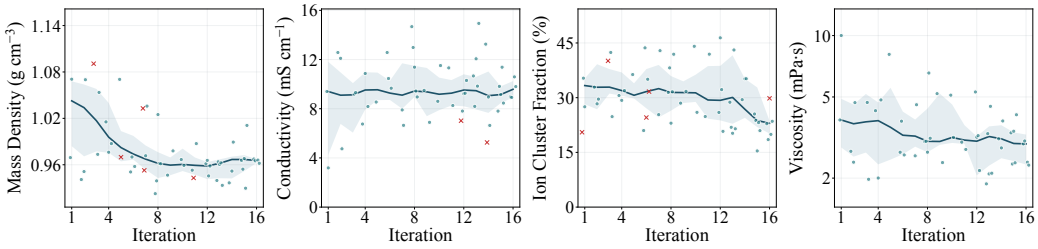



Figure 10: Self-improvement trace for the low-temperature setting.

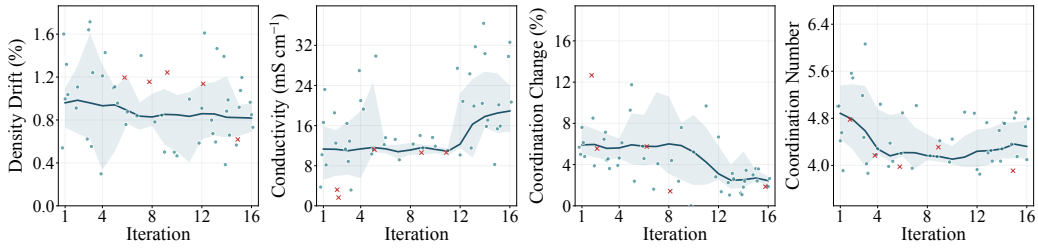



Figure 11: Self-improvement trace for the high-temperature setting.

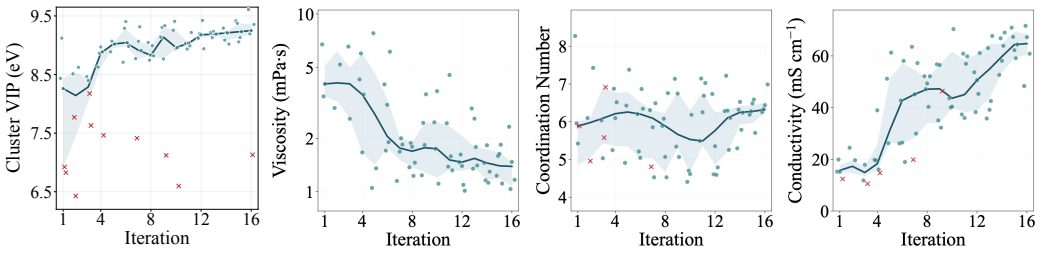



Figure 12: Self-improvement trace for the high-voltage setting.

**Convergence of Three MD Trajectories.** Figures 13 to 16, together with Figure 3, provide MD convergence visualizations for physical evaluation across tasks. These results show that SAGE's physical evaluation can detect whether constrained properties converge to the desired values. Across settings, most observables show rapid transient behavior followed by stable plateaus, indicating that the simulation horizon is sufficient to support downstream constraint assessment. The low- and high-temperature cases exhibit stronger early fluctuations, but their trajectories still become comparatively steady over time, highlighting the importance of explicit physics-driven validation near more challenging feasible boundaries. Note that the values are z-score normalized.

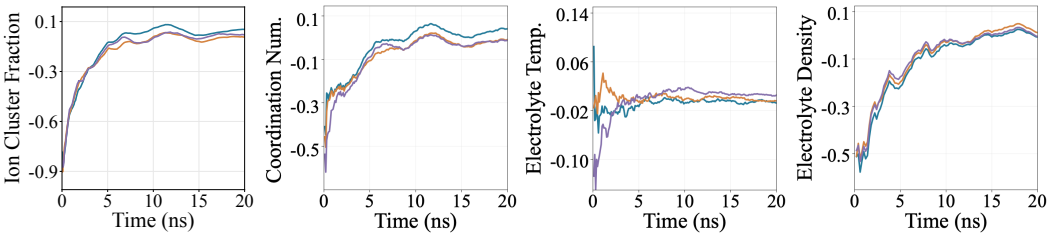



Figure 13: Convergence of MD trajectories under physical validation for the fast-charge setting.

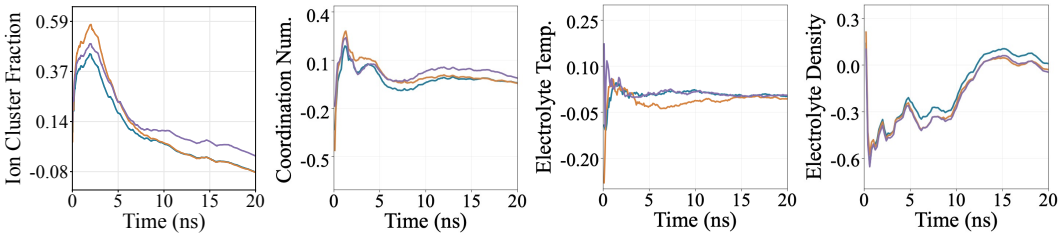



Figure 14: Convergence of MD trajectories under physical validation for the low-temperature setting.

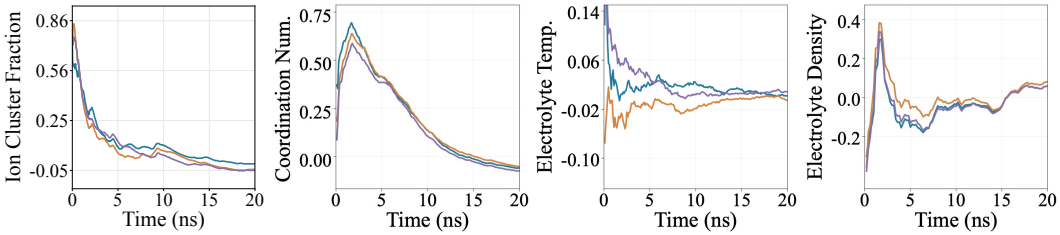



Figure 15: Convergence of MD trajectories under physical validation for the high-temperature setting.

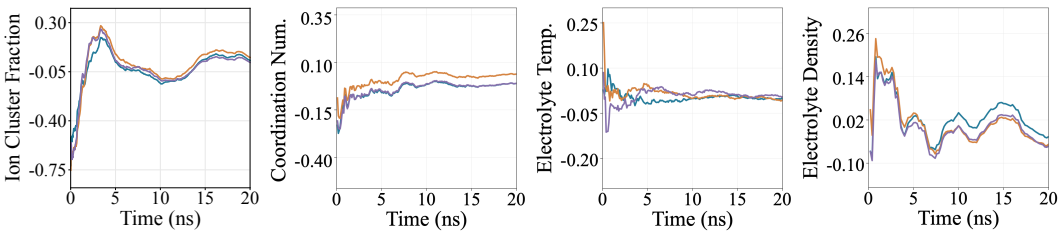



Figure 16: Convergence of MD trajectories under physical validation for the high-voltage setting.

**Performance Gain from the SAGE Workflow.** Figures 17 to 20, together with Figure 5, show that SAGE substantially improves base-model performance by more than 40 percentage points on average, highlighting the importance of strategic self-improvement through physics-driven evaluation. The gains are consistent across model families, including both stronger and weaker backbones, which suggests that the improvement comes from the workflow rather than from model scale alone. This result indicates that structured physical feedback and iterative policy refinement can unlock remarkable formulation performance that fixed-prompt generation leaves unrealized.

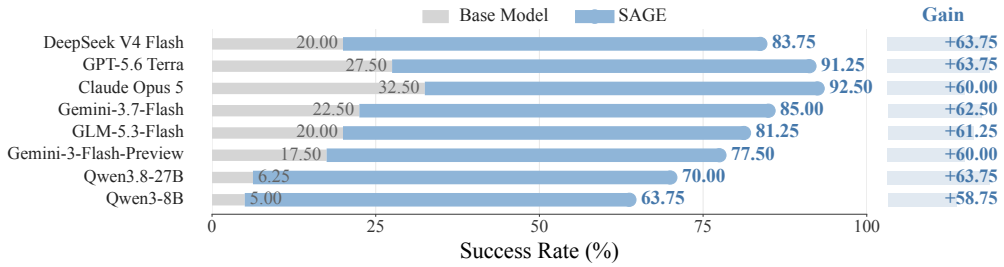



Figure 17: Performance gain from SAGE workflow for the fast-charge setting.

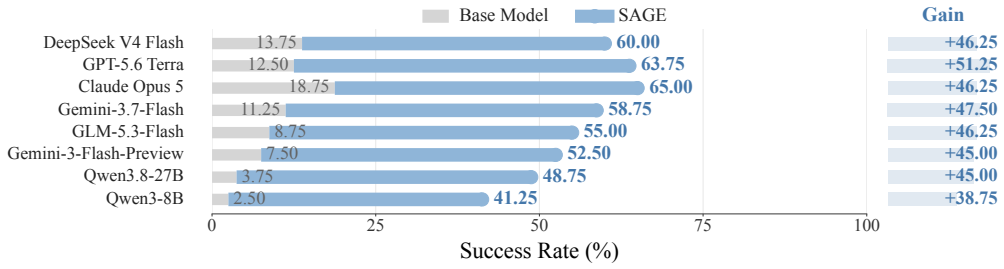



Figure 18: Performance gain from SAGE workflow for the low-temperature setting.

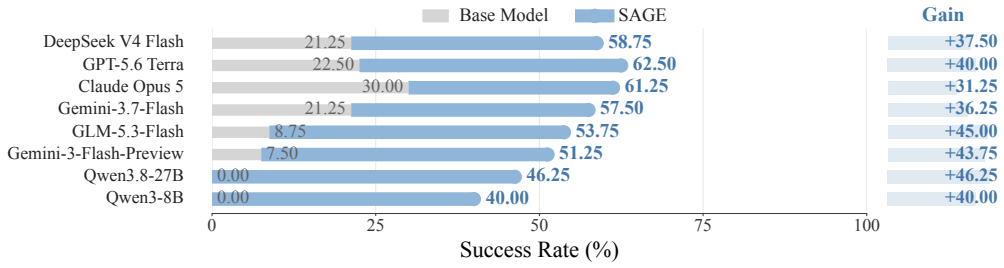



Figure 19: Performance gain from SAGE workflow for the high-temperature setting.

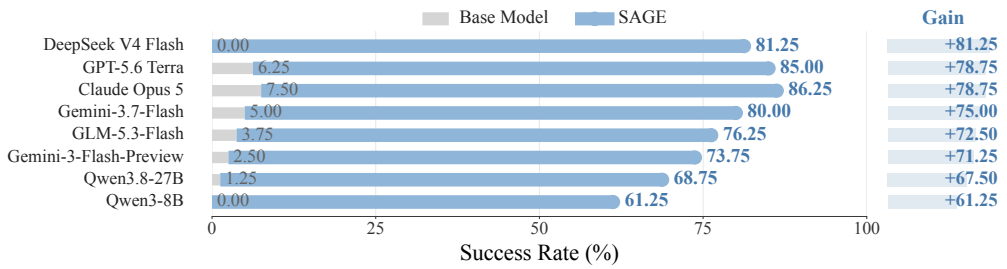



Figure 20: Performance gain from SAGE workflow for the high-voltage setting.

**Property Distributions under MD Validation.** Figures 21 to 24, together with Figure 4, show the electrochemical properties for the generated formulations. The distributions reveal clear structured regions rather than diffuse coverage, suggesting that SAGE does not sample arbitrarily but instead concentrates search in physicochemically plausible parts of the formulation space. They also highlight persistent trade-offs among conductivity, viscosity, coordination, and clustering, which helps explain why viable formulations emerge only when multiple constraints are balanced jointly. Here, the density on the right-hand side of the figure refers to *kernel density*, and the figure visualizes all designed formulations, including both Pass and Fail cases.

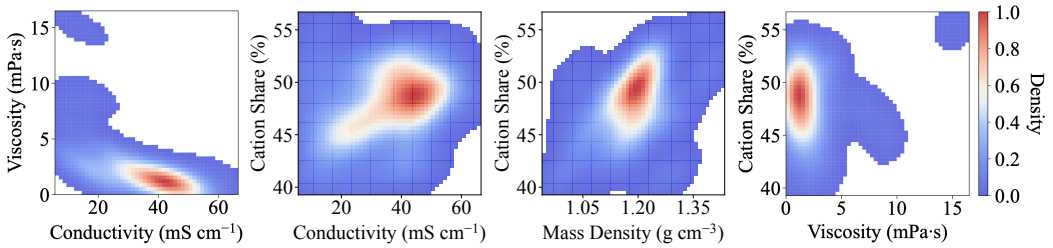



Figure 21: Key electrochemical properties obtained from MD simulations of designed electrolyte formulations in the fast-charge setting using an OPLS force field (Jorgensen et al., 1996).

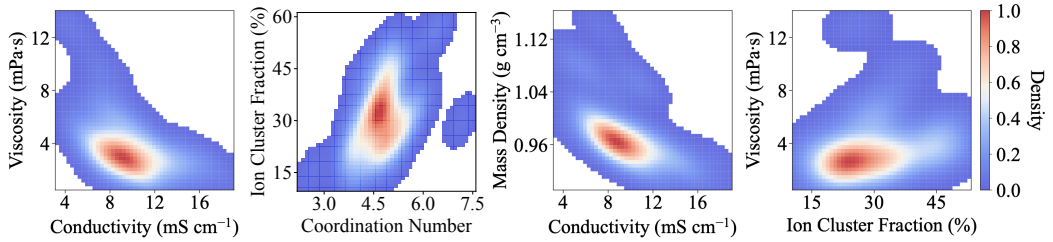



Figure 22: Key electrochemical properties obtained from MD simulations of designed electrolyte formulations in the low-temperature setting using an OPLS force field (Jorgensen et al., 1996).

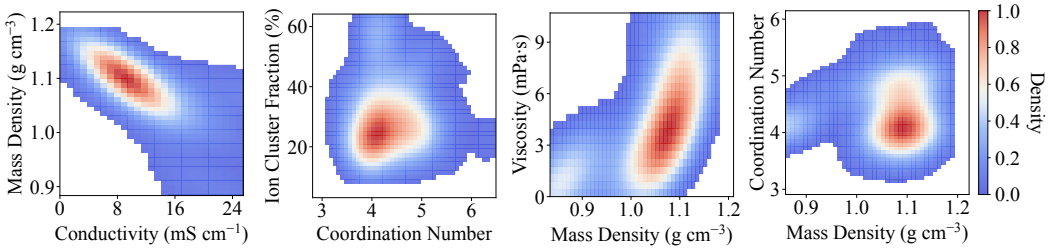



Figure 23: Key electrochemical properties obtained from MD simulations of designed electrolyte formulations in the high-temperature setting using an OPLS force field (Jorgensen et al., 1996).

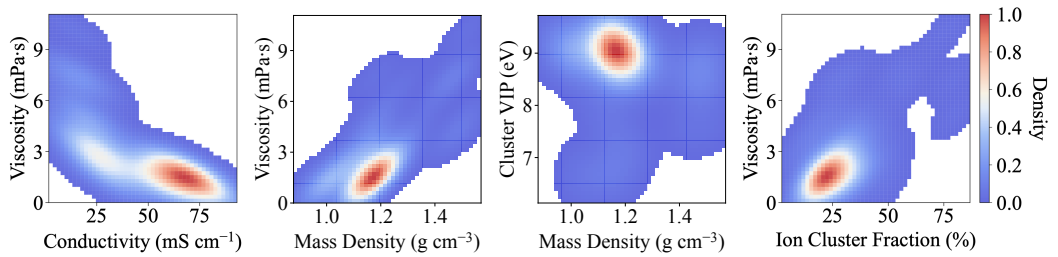



Figure 24: Key electrochemical properties of designed high-voltage electrolyte formulations, obtained from MD simulations and DFT calculations using the OPLS force field (Jorgensen et al., 1996).

**Structured Artifacts for Performance Improvement.** Figures 25 and 26, together with Figure 6, visualize the performance gains obtained by reusing structured knowledge from the final iteration in fresh runs of the same task. This structured knowledge consists of update logs for agent skills, task prompts, and tool-use policies, namely the accumulated design policy updates. At the reported evaluation points, the results show that transferring these final-iteration artifacts improves the design success rate relative to control runs without the updated design policy. This pattern indicates that SAGE learns and distills useful experience into an explicit and reusable form rather than relying only on transient search history. It also suggests that self-improvement produces portable task-level knowledge that can accelerate later discovery under the same constraints. The observed gains across tasks further suggest that these artifacts capture practically effective guidance rather than task-specific noise. For this study, we report aggregate success rates across runs to reduce the high variance of five-candidate batches. Specifically, we evaluate the top 10 candidates per iteration and repeat the experiment independently five times.

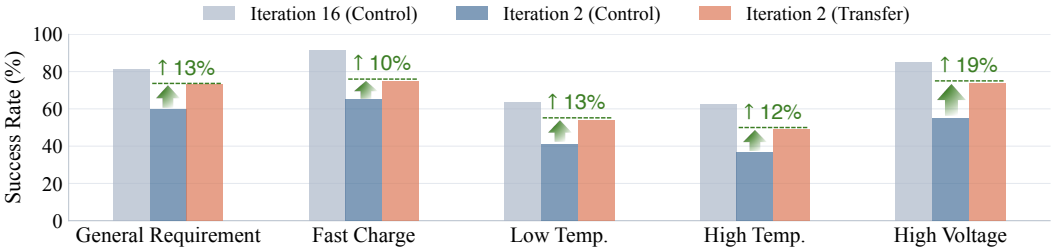



Figure 25: Transfer of structured design policy artifacts improves success rates on the same task for the GPT-5.6 backbone (Singh et al., 2025).

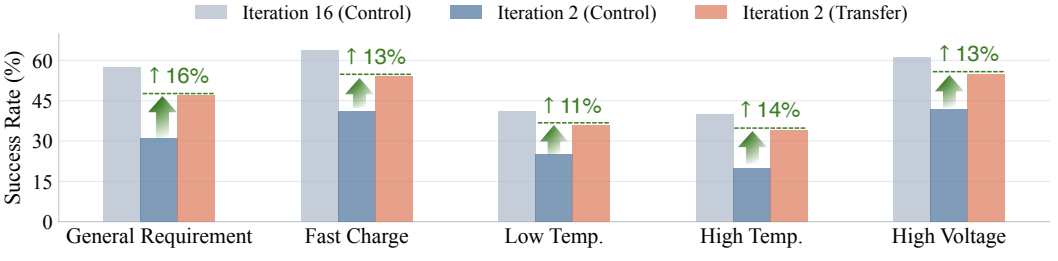



Figure 26: Transfer of structured design policy artifacts improves success rates on the same task for the Qwen3-8B backbone (Yang et al., 2025).

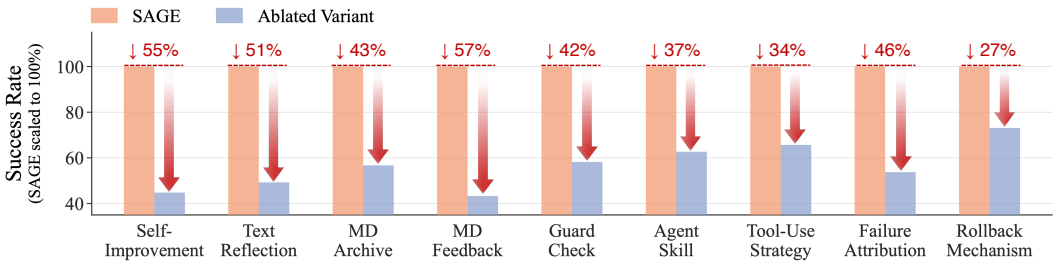



Figure 27: Removing or replacing core components changes performance in the fast-charge setting, revealing the contribution of iterative self-improvement and physics-driven feedback.

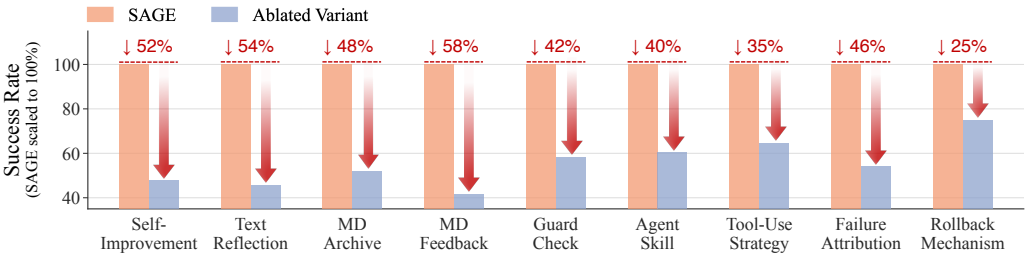



Figure 28: Removing or replacing core components changes performance in the low-temperature setting, revealing the contribution of iterative self-improvement and physics-driven feedback.

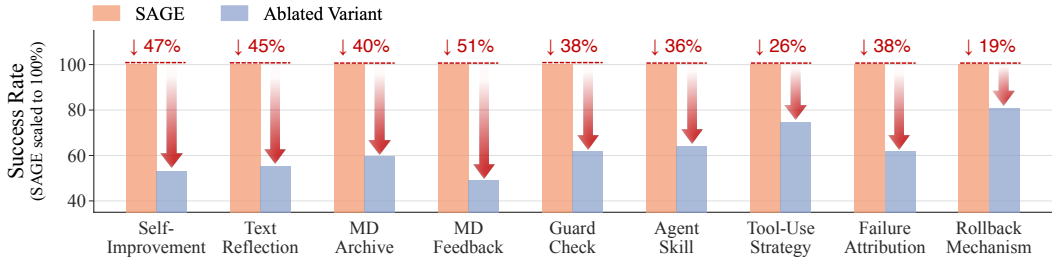



Figure 29: Removing or replacing core components changes performance in the high-temperature setting, revealing the contribution of iterative self-improvement and physics-driven feedback.

**Ablation Studies.** Figures 27 to 29, together with Figure 7, showcase how restricting parts of the feedback and policy stack changes success rates. Across tasks, every ablated variant reduces performance, confirming that SAGE's gains do not come from a single isolated component. The largest drops are typically observed when MD feedback and self-improvement are removed, underscoring the central role of physics-grounded feedback and explicit policy updating. By contrast, guard checks, agent skills, tool-use strategies, failure attribution, and rollback mechanisms also contribute meaningfully, even when their individual removals produce smaller declines. This pattern suggests that SAGE works as an integrated system in which reliable improvement depends on both informative physical signals and the structured mechanisms that convert those signals into better search behavior. The consistent degradation across settings further indicates that these components remain important under different task constraints rather than only in a single optimization regime. It also demonstrates that robust electrolyte design requires both effective candidate generation and a dependable self-correction mechanism when proposals fail to satisfy coupled constraints. Note that SAGE is scaled to 100%, and ablated variants are reported relative to this baseline.

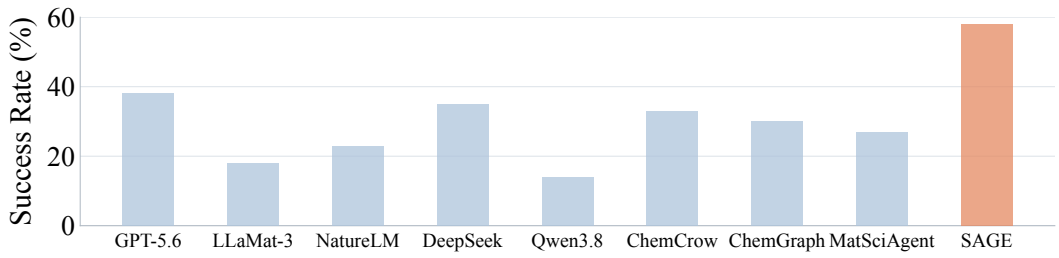



Figure 30: Success rates for directed formulation editing across all baselines on the general requirements task. SAGE achieves the highest performance, indicating a stronger ability to make local, physically valid formulation edits that improve the target property while preserving task constraints.

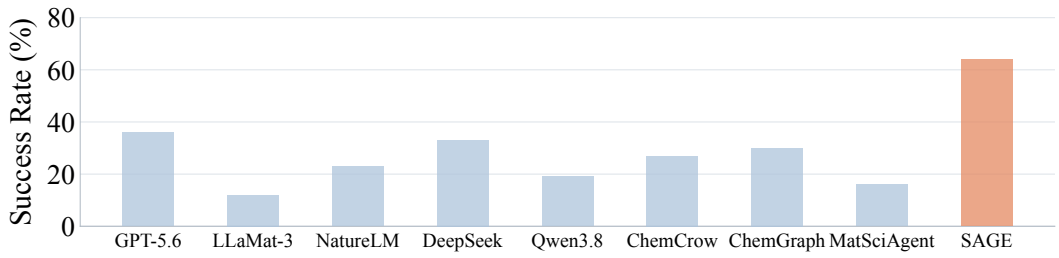



Figure 31: Success rates for directed formulation editing across all baselines on the fast-charge task. SAGE achieves the highest performance, indicating a stronger ability to make local, physically valid formulation edits that improve the target property while preserving task constraints.

**Directed Formulation Editing.** As shown in Figures 30 and 31, we evaluate whether SAGE can perform directed formulation editing, a stricter setting than open-ended formulation discovery. This setting is practically important because electrolyte optimization often begins from an existing feasible formulation and requires targeted, minimal edits rather than *de novo* generation. Specifically, each method is asked to edit 100 shared, frozen valid reference formulations per task. For each reference, one proposal is generated for each of two beneficial directions, while salt and additive identities and amounts are held fixed and only solvent ratios or at most one solvent identity may be changed, subject to a solvent-composition $L1$ distance of 0.02–0.40. Each proposal receives a single physical evaluation, and success requires valid QC, a strict change in the specified direction, and continued satisfaction of all task-specific constraints. Thus, the task tests whether a method can make a local, chemically valid edit that moves a target property in a prescribed direction while preserving feasibility. Because salt identity, additive identity, and edit budget are fixed, the experiment evaluates intervention quality rather than memorization or unconstrained generation. All methods share the same deterministic programs, scientific tools, and physical evaluation protocols, so the comparison primarily reflects differences in retrieval, reasoning, and edit proposal. The results show that SAGE achieves the highest success rate among all compared methods, with a clear margin over both strong single-agent baselines and other agentic workflows. Its advantage on both tasks indicates that the benefits of SAGE extend beyond open-ended search to controllable formulation refinement under strict feasibility requirements. More broadly, these results suggest that SAGE's constraint-aware proposal process and physics-guided design policy refinement improve its ability to make precise, physically feasible formulation edits under tightly coupled scientific constraints.

### A.8 3D Visualization of Designed Formulations

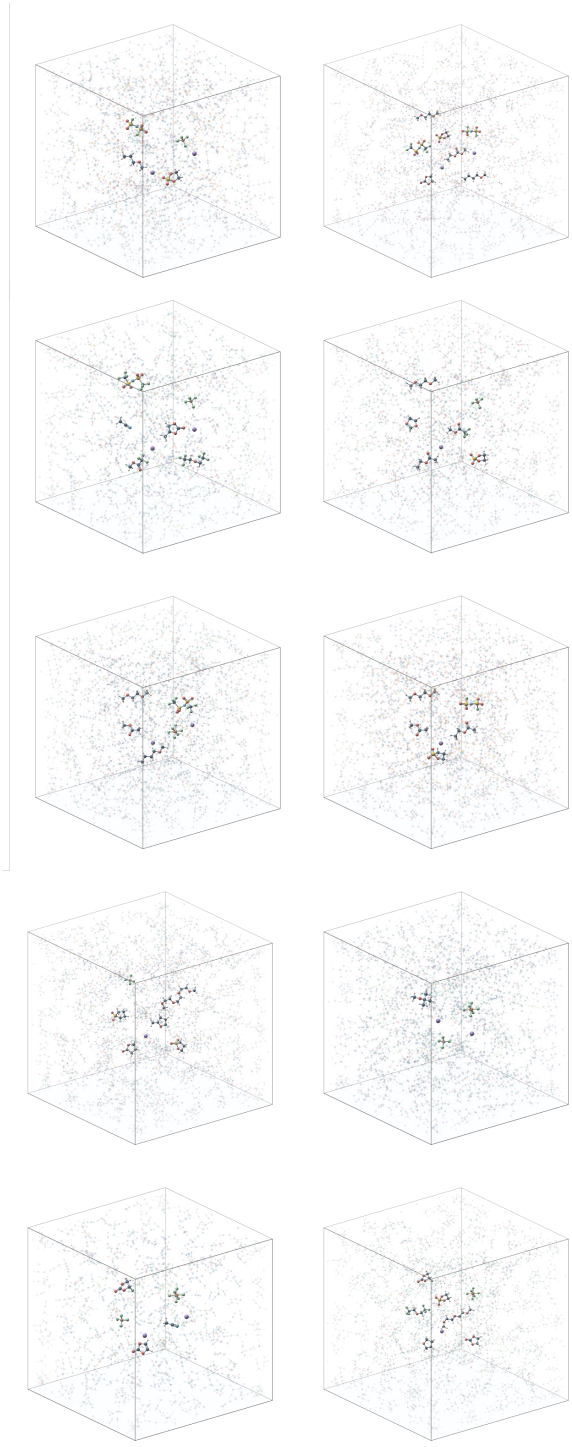


Figure 32: Representative 3D atomistic snapshots of MD simulation cells for designed formulations in tasks 1–5 (top to bottom), with two randomly selected examples shown in each row. Colored markers highlight the primary solvation and coordination structures, while the remaining grey markers denote the surrounding molecules in the periodic simulation box.

### A.9 Artifacts from the Final Iteration Improve Task Performance

Instead of restarting task 5 from scratch, we reinitialize the agent with structured knowledge accumulated during the original task 5 run, including the skill library, the task-specific prompt, and the tool-use strategy consolidated by that stage. This experiment studies an intra-task warm start. Its purpose is to evaluate policy persistence and the reduction of cold-start cost under the same task specification, component pool, physical protocol, and gate definition. In other words, the task itself is unchanged, and only the initial agent state differs. This makes the setting useful for testing whether the agent can carry forward a reusable search policy rather than merely reproduce isolated successful candidates. The central premise is that, if these transferred components already encode a useful approximation of task 5's stability boundary in composition space, then the agent should not need to spend many additional iterations rediscovering the same pattern. Our results are consistent with this expectation. When task 5 is rerun from the iteration 16 state, the new iteration 2 already reaches a pass rate and formulation quality that previously required substantially more iterations in the original run. This pattern is consistent with the benefit of the accumulated skill, prompt, and tool-use state, and suggests that a meaningful part of the original search cost lay in locating a productive region rather than refining it once found. At the same time, this result does not isolate the contribution of any single artifact, nor does it establish transfer across different task regimes. We present the artifacts from iteration 16, which is the final iteration of the original run, below as logs of design-policy updates. The fact that a strong pass rate emerges as early as iteration 2 in the rerun, rather than only after many iterations as in the original search, is consistent with a reduction in the cold-start cost that typically dominates the early stage of exploration. Note that, by design, the transferred documents contain only abstract policy-level knowledge and do not include historical formulations, pass labels or answers, exact ratios, evaluation cache, full dialogue histories, or the task archive.

From an agentic perspective, the iteration 16 skill library is not a single hypothesis, but an evidence-weighted policy over search operators such as `probe_failed_gate_boundary`, `mutate_ratio`, and `seed_from_low_discrepancy_design`. These operators are associated with the task's recorded counterexamples and confirmed failures. Reusing this policy allows the proposer to reduce the early trial-and-error phase that would otherwise be needed to identify useful operators for task 5. Instead, it starts with a heuristic bias toward carbonate-majority, low-ether compositions and toward probing near the gate boundary rather than sampling broadly. This bias is treated as a search prior to be tested, not as an established chemical rule. This is analogous to warm-starting an optimizer from a previously prioritized region instead of a random initialization. As a result, more of the available search budget can be devoted to refining that region rather than locating it from scratch. The transferred prompt from iteration 16 further preserves alignment between the skill library and the proposal policy. The tool-use strategy plays a similar role for retrieval. In particular, its `candidate_seeding_bias` guides the first batch toward regions prioritized by the recorded policy. In the rerun, iteration 1 achieves an external pass rate of 4/5. Taken together, these observations are consistent with reduced cold-start cost under the same task specification, but they do not isolate the effect of the chemical seeding bias from other transferred state. They also do not establish cross-task transfer or a universal component rule.

From a scientific perspective, the transferred state encodes a task-specific working hypothesis derived from repeated $\Delta$SCF DFT feedback. The labels "carbonate-majority" and "low-ether" describe the policy's operational seeding categories rather than universal solvent classes, and the available records do not provide a controlled comparison sufficient to establish a general relative trend. Successful formulations outside these categories are retained as counterexamples. The rerun therefore begins with a search prior shaped by prior physical evaluations under the same gate, rather than a validated mechanistic rule. The rapid convergence observed by iteration 2 should be interpreted as evidence consistent with policy persistence within task 5 and reduced cold-start cost within the same regime, rather than as evidence of causal chemical attribution or transfer across different constraint settings. Below, we provide an illustrative excerpt from a randomly selected run that documents the task 5 artifacts in the form of logs.

**Task 5 skill library snapshot (end of R16, carried forward)**

```
skill_library_view:
  task: T5
  snapshot_iteration: 16
  active_skills:
  - skill_id: s_t5_r01_ffb9ba
    hypothesis_tag: ether_rich_lowers_cluster_vip
    operator_weights: {mutate_ratio: 2.0, probe_failed_gate_boundary: 2.0}
    status: retained
  - skill_id: s_t5_r02_493c12
    hypothesis_tag: carbonate_majority_favors_pass
    operator_weights: {mutate_ratio: 2.0, seed_from_low_discrepancy_design: 2.0}
    status: retained
  - skill_id: s_t5_r09_f4852e
    hypothesis_tag: local_solvent_salt_neighborhood_below_gate
    operator_weights: {probe_failed_gate_boundary: 2.0, mutate_ratio: 2.0}
    status: retained
    focus_gate: T5_DFT_VIP
  consolidated_operator_priority:
  - probe_failed_gate_boundary
  - mutate_ratio
  - seed_from_low_discrepancy_design
  evidence_focus_gates: [T5_DFT_VIP]
```

Source: Export/R16/04_skill_library_export.md

**Task 5 task-specific prompt view (end of R16, carried forward)**

```
task_prompt_view:
  task: T5
  active_content_hash: 7a2e18f0...
  proposer_constraints:
  - draw components only from the registered T5 allowed pool
  - respect the mutate_ratio and probe_failed_gate_boundary operator
    contracts registered in the skill library
  - do not invent property values. Defer pass/fail authority to
    measured MD/DFT and QC
  reflector_constraints:
  - anchor every new hypothesis to the current evidence_focus_gates
  - update hypothesis_tag rather than discard prior skill entries
    unless directly contradicted by new failed cases
  behavior_changed_since_R01: true
```

Source: Export/R16/06_prompt_export.md

**Task 5 tool-use strategy view (end of R16, carried forward)**

```
tool_use_strategy_view:
  task: T5
  retrieval_gate_priority:
  - T5_DFT_VIP
  external_validation_top_k: 5
  candidate_seeding_bias: >
    prioritize carbonate-majority, low-ether-fraction compositions
    near previously confirmed passing neighborhoods before exploring
    ether-rich regions
  behavior_changed_since_R02: true
```

Source: Export/R16/07_tool_use_policy_export.md

**Task 5 consolidated hypothesis and evidence pool (end of R16)**

```
hypothesis_evidence_pool:
  task: T5
  standing_hypothesis: >
```

```
    Ether-linkage-rich solvent blends tend to lower the
    anion-solvent cluster VIP relative to carbonate-majority blends.
    The effect is treated as graded with ether fraction rather than binary.
  supporting_failed_cases: 5
  provenance:
  - history://T5/failure/case_09b51c6f6bf787d8
  - history://T5/failure/case_e5ba40f112a8a69d
  - history://T5/failure/case_ac11d321fe753138
  - history://T5/failure/case_dda9d02f09efe70d
  - history://T5/failure/case_a81c5cdbf13c626a
  attribution_flag: not_causal_attribution
  numeric_authority_flag: llm_numeric_authority_false
```

Source: Export/R16/08_evidence_pool_export.md

**Task 5 constraint check (end of R16, carried forward)**

```
constraint_check_result:
  task: T5
  catalog_id_check: pass
  locked_threshold_check: pass
  operator_whitelist_check: pass
  patch_reversibility_check: pass
  evidence_backing_check: pass
  overall_status: compliant
```

Source: Export/R16/05_constraint_check.md

**Task 5 re-run, Iteration 1: warm-started from the original R16 state**

```
iteration_summary:
  run_id: Task5-rerun-warmstart
  iteration: 1
  run_type: illustrative_re_run
  warm_start_source: original_R16_state
  seeding_source: consolidated_operator_priority + candidate_seeding_bias
  reporting_scope: >
    Illustrative qualitative trace excerpts; this re-run is not used to derive
    the aggregate iteration-level statistics reported in the results.
  note: >
    Candidates were generated using the carbonate-majority,
    low-ether-fraction seeding bias. The standing hypothesis was
    applied without introducing a new hypothesis.
```

Source: Export/ReRun_Task5/R01/01_iteration_summary.md

**Task 5 re-run, Iteration 2**

```
iteration_summary:
  run_id: Task5-rerun-warmstart
  iteration: 2
  run_type: illustrative_re_run
  reporting_scope: >
    Illustrative qualitative trace excerpts; this re-run is not used to derive the
     aggregate iteration-level statistics reported in the results.
  patch_decision:
    status: no_patch_needed
    reason: >
      The standing hypothesis and consolidated operator priority were
      retained without modification.
```

Source: Export/ReRun_Task5/R02/01_iteration_summary.md

### A.10 Case Study: High-Voltage Battery Formulation Design

**Case Study Setup.** In this section, we use task 5, high-voltage electrolyte formulation design, as a case study to illustrate how iterative self-improvement improves formulation design through structured updates to key components, including agent skills, task prompts, tool-use strategies, hypotheses, and supporting evidence. We randomly select one experimental run and present its update logs and results as an illustrative example. Specifically, task 5 searches formulations with one or more lithium salts, two to five solvents, at most three additives, and no more than seven total components. The final candidate protocol combines a finite-time Nernst–Einstein transport screen with density, coordination, persistent-cluster, viscosity, quality-control, and cluster VIP gates. In this case, `T5_DFT_VIP` is the physical gate that remains the dominant bottleneck across the four feedback windows. The VIP acceptance criterion is fixed throughout the task and does not change across iterations. The matched-protocol reference cluster is fixed before R01 under the task 5 cluster-construction and $\Delta$SCF procedure and is reused for every method and feedback window. The case study therefore does not treat an iteration-specific telemetry field as a separate acceptance criterion.

This case is designed to expose the decision flow of the method, not to claim that a single solvent is universally better. Each iteration begins with external MD or $\Delta$SCF feedback and ends with a new batch of numerically instantiated candidates. Between those endpoints, task memory records candidate-level facts, the reflector classifies failures and proposes bounded policy changes, the agent skill registry stores reusable experiences, the retriever updates its query context, the evidence compiler constructs a source-linked evidence bundle, the prompt context carries active skills and local archive state, the proposer emits typed qualitative hypotheses, and the deterministic composer assigns concentrations and ratios. The external evaluator supplies physical measurements and final PASS or FAIL labels. A numerical instability remains a process issue, an uncertainty-capped result remains uncertainty-capped, and neither is rewritten as a chemical threshold miss. Below, we provide an illustrative excerpt from a randomly selected run as a case study.

**Case-study contract from the prompt and task-memory records**

```
Proposer role:
  Form qualitative interaction hypotheses. Numerical composition is generated by the typed
      composer.
  Use registered component IDs, frozen gate IDs, supplied evidence IDs and permitted
     operators.
  Never output fractions, concentrations, thresholds, invented property values, PASS
     labels,
  SMILES, or invented IDs.
  Do not globally ban a component from one failure. Reason about salt-solvent-
     concentration
  interactions and retain pass counterexamples.

Authority boundary:
  skill and prompt changes alter search context
  typed composer generates numerical recipes
  measured MD/DFT and QC assign physical outcomes
```

Source: Update/U1_after_R01/06_prompt.md and case study/task 5 case study.md

The quoted contract is not merely a formatting convention. It safeguards the definition of success throughout the discovery loop. The task specification, candidate gating criteria, and physical evaluator remain fixed, while only the search distribution is allowed to change. This separation ensures that the subsequent improvements can be attributed to structured, failure-directed search rather than to relaxed thresholds or unverified model self-assessment.

For the archive fields below, `entries` denotes serialized records present in the bounded prompt context, including metadata-only parent or provenance records. `stored_status_counts` describes the status composition of that context snapshot. It is not a cumulative counter of physical evaluations. The cumulative external outcomes are reported separately in the case-study narrative.

**Iteration U1, R01 to R02: making the first physical failures actionable.** The first iteration submitted five candidates to external validation. Three passed and two failed the VIP gate with quality-controlled measurements. Candidate `T5_r01_0017_e6f8f8b8` had a VIP of

6.9232856 eV, a pass probability of 0.01, and a valid transport value of $8.3197\,\mathrm{mS\,cm^{-1}}$. Candidate `T5_r01_0018_130736ae` had a VIP of 6.8267733 eV, a pass probability of 0.0, and a valid transport value of $10.1347\,\mathrm{mS\,cm^{-1}}$. The task-memory component retained both failures, their result identifiers, measured values, QC status, and failure references. It also retained the three PASS records, because a component appearing in a failed mixture can also appear in a successful mixture and therefore cannot be removed globally from the search space.

**U1 task memory: the physical facts handed to reflection**

```
my_completed_iteration: 1
my_recorded_counts:
  generated: 20
  external_submitted: 5
  new_valid: 3
  valid_pool_total: 3
  external_pass: 3
  external_failed: 2

- candidate_id: T5_r01_0017_e6f8f8b8
  result_id: mdr_e6fc0ef82b87b648cc39
  status: FAILED
  failure_codes: [THRESHOLD_MISS]
  vip_eV: 6.923285599951515
  vip_ci90_eV: [6.695745665000114, 7.442749382670243]
  vip_qc: true
  vip_pass_probability: 0.01
  NE_mobility_mS_cm: 8.31965453726443

- candidate_id: T5_r01_0018_130736ae
  result_id: mdr_96b5ad3846de35ba0004
  status: FAILED
  failure_codes: [THRESHOLD_MISS]
  vip_eV: 6.826773261057117
  vip_qc: true
  vip_pass_probability: 0.0
  NE_mobility_mS_cm: 10.134718880848311
```

Source: Update/U1_after_R01/01_task_memory.md

The reflector did not infer a universal ether penalty from these two records. Instead, it proposed investigating local ratio changes and failed-gate boundary probes while retaining successful counterexamples. The typed intent focused on `T5_DFT_VIP` and recommended `mutate_ratio` and `probe_failed_gate_boundary`. The rollback check was not eligible because this was the root policy update, so the patch was committed as `p_T5_r01_01` at policy version 1.

**U1 reflection and canonical skill registration**

```
reflection_update:
  task: T5
  iteration: 1
  focus_gate_ids: [T5_DFT_VIP]
  recommended_operators:
  - mutate_ratio
  - probe_failed_gate_boundary

registered_skill:
  skill_id: s_t5_r01_ffb9ba
  created_iteration: 1
  applicable_tasks: [T5]
  failure_refs:
  - history://T5/failure/f_79a75f21ce764641
  - history://T5/failure/f_a291bf8f00fd88c6
  machine_prior:
    operator_weights:
      mutate_ratio: 2.0
      probe_failed_gate_boundary: 2.0
    evidence_focus_gates: [T5_DFT_VIP]
```

```
    failed_cases_in_next_request: 2
    pass_counterexamples_in_next_request: 3
  patch_id: p_t5_r01_01
```

Source: Update/U1_after_R01/02_reflection.md and 03_skill.md

This skill is more precise than a statement such as "prefer high-VIP solvents." It is triggered by measured VIP deficits in a specific salt–solvent neighborhood, stores the exact failure references, and contributes two machine-readable operator weights. These weights are preferences rather than execution claims. The later composer record is therefore needed to determine which operators actually generated candidates.

The retriever received the two observed failure signatures before retrieving R02 evidence. Its tool policy was still at version 0 and had no persistent retrieval-gate priority, but the query was already more specific than the original prospective task query. The executed call used the task-isolated knowledge retriever with four source classes and returned a receipt containing input and output hashes. The evidence compiler then produced 84 cards, 81 from literature retrieval and three from user or task evidence. One selected card discussed solvent–salt charge-transfer complexes. The compiler explicitly marked this card as contextual, without an extracted protocol or assessed uncertainty.

**U1 tool use and evidence bundle**

```
policy_I_received_before:
  external_validation_top_k: 5
  retrieval_source_weights:
    dataset_molecules: 0.4
    literature_rag: 0.45
    pubchem_props: 0.15
    user_evidence: 0.6
  retrieval_gate_priority: []

retrieval_call_I_executed:
  receipt_id: tool_4e1b7df622ed07fe
  tool: knowledge_retriever
  version: knowledge_retriever_v2_task_isolated
  status: ok
  query: electrolyte solvent anion cluster VIP charge transfer
  task_id: T5
  top_k: 24

my_updated_evidence_state:
  artifact_id: a_a7a7d2489cbf9874
  cards: 84
  source_type_counts: {literature_rag: 81, user_evidence: 3}
  critical_evidence_refs:
  - ev_eb552160d08b73d3dd7d
  - ev_44b67b490e116fb789c4
  - ev_feed84cd6d766508a5a7
  selected_contextual_card:
    evidence_id: ev_cee27af3f3b35a19fc6f
    direct_to_failed_gate: false
    protocol: not_extracted
    uncertainty: not_assessed
```

Source: Update/U1_after_R01/04_tool_use.md and 05_evidence_update.md

The prompt context for R02 changed even though the fixed role instruction did not. The request now contained one active skill, two observed VIP failures with failure references, five local results carrying metrics, and the evidence bundle. The prompt audit reported 47,699 estimated tokens against a 48,000-token budget and retained all hard fields, including allowed components, failure evidence, local archive, skills, and task specification. The proposer therefore moved from generic prospective targets to a local comparison between measured failures and measured PASS counterexamples. It still returned only qualitative hypotheses and operator recommendations.

**U1 prompt context and composer handoff**

```
my_current_context:
  active_skill_ids: [s_t5_r01_ffb9ba]
  failure_signatures: 2 observed T5_DFT_VIP failures
  archive_context:
    entries: 5
    entries_with_metrics: 5
    stored_status_counts: {FAILED: 2, PASS: 3}
    entry_count_is_not_independent_evaluation_count: true

my_effective_operator_weights:
  seed_from_evidence: 2.0
  mutate_ratio: 5.0
  probe_failed_gate_boundary: 5.0
  swap_component: 2.0
  recombine_compatible_parents: 1.0
  split_primary_salt: 1.0
operators_in_my_generated_batch:
  archive_ratio_or_component_mutation: 10
  split_primary_salt: 6
  seed_from_low_discrepancy_design: 4
my_batch_allocation:
  repair: 16
  diversity_control: 4
```

Source: Update/U1_after_R01/06_prompt.md and 08_compose_update.md

The composer generated 20 R02 candidates and sent five selected candidates to external validation. The key change was not that the model directly proposed a better recipe. Rather, the first physical failures were compiled into a task-specific skill, a failure-focused query, a source-linked evidence bundle, and a prompt with measured counterexamples. The numerical composer then converted these qualitative directions into admissible candidates under unchanged task constraints.

**Iteration U2, R02 to R03: preserving the distinction between uncertainty and failure.** R02 produced two PASS results and three failures, but the three failures carried different meanings. Candidate `T5_r02_0025_949a38c5` returned `UNCERTAIN_AT_CAP`. The evaluator stopped before producing a finalized VIP probability under the frozen gate. Candidate `T5_r02_0079_4027b930` had VIP 6.4285682 eV and was a clear `THRESHOLD_MISS`. Candidate `T5_r02_0083_540fde7b` was labeled `NUMERICAL_INSTABILITY` and had no VIP value. Task memory preserved the capped verdict separately from the clear threshold miss. The uncertainty-capped result was not promoted to PASS or counted as a conclusive chemical threshold failure, and the numerical failure was not treated as evidence of low oxidation stability. The provisional numeric fields associated with the capped result are not used as authoritative measurements in this case study.

**U2 task memory: three failure semantics and updated pool**

```
my_completed_iteration: 2
my_recorded_counts:
  external_submitted: 5
  new_valid: 2
  valid_pool_total: 5
  external_pass: 2
  external_failed: 3

- candidate_id: T5_r02_0025_949a38c5
  status: FAILED
  failure_codes: [UNCERTAIN_AT_CAP]
  NE_mobility_mS_cm: 16.029272475238212

- candidate_id: T5_r02_0079_4027b930
  status: FAILED
  failure_codes: [THRESHOLD_MISS]
  vip_eV: 6.42856818355342
  vip_pass_probability: 0.0
  NE_mobility_mS_cm: 15.982716408915216

- candidate_id: T5_r02_0083_540fde7b
  status: FAILED
```

```
  failure_codes: [NUMERICAL_INSTABILITY]
  vip_eV: null
  vip_pass_probability: null
```

Source: Update/U2_after_R02/01_task_memory.md

The reflector produced two corresponding search responses. It proposed local ratio mutation around the uncertain candidate and low-discrepancy seeds for broader coverage. Its rollback check was eligible because a previous increment existed, but it returned `should_rollback: false`. Under the rollback definition, the denominator remains the five physically evaluated candidates. The numerator of $f_k$ counts only clear, quality-controlled instances of the targeted chemical failure mode. `UNCERTAIN_AT_CAP` is unresolved rather than a conclusive threshold miss, and `NUMERICAL_INSTABILITY` is excluded as an execution-only failure. Thus the clear VIP-miss count fell from two in R01 to one in R02, so the targeted failure rate did not worsen even though the valid increment dropped from three to two. The policy compiler committed patches `p_T5_r02_01` and `p_T5_r02_02` at policy version 3.

The second skill, `s_t5_r02_493c12`, was appended to the first skill rather than replacing it. Its natural-language lesson explicitly distinguished a promising but uncertain point from a clear deficit. Its machine prior assigned weight 2.0 to `mutate_ratio` and 2.0 to `seed_from_low_discrepancy_design`, kept `T5_DFT_VIP` as the evidence focus, and passed two failure references and seven evidence references into the next context. The active skill list therefore encoded cumulative memory rather than last-iteration overwrite.

**U2 skill accumulation and canonical policy registration**

```
skill_registration_update:
  task: T5
  iteration: 2
  active_skill_ids_before:
  - s_t5_r01_ffb9ba
  active_skill_ids_after:
  - s_t5_r01_ffb9ba
  - s_t5_r02_493c12

registered_skill:
  skill_id: s_t5_r02_493c12
  created_iteration: 2
  applicable_tasks: [T5]
  failure_refs:
  - history://T5/failure/f_a5317237a12b0205
  - history://T5/failure/f_3b6b717a5a44a65f
  machine_prior:
    operator_weights:
      mutate_ratio: 2.0
      seed_from_low_discrepancy_design: 2.0
    evidence_focus_gates: [T5_DFT_VIP]
    failed_cases_in_next_request: 2
    pass_counterexamples_in_next_request: 5
  patch_id: p_t5_r02_01
```

Source: Update/U2_after_R02/02_reflection.md and 03_skill.md

At the same boundary, tool use changed from a contextual query to a persistent gate-prioritized query. The policy changed `/retrieval_gate_priority` from an empty list to `[T5_DFT_VIP]`. The retriever preserved its source weights, top-five external-validation allocation, and task isolation. It also added queries about $LiBF_4$ cluster ionization, VC-related high-voltage evidence, and solvent-cluster oxidation windows. The record distinguishes "priority applied" from "priority committed," which prevents a policy acknowledgement from being mistaken for an executed retrieval effect.

**U2 tool use and evidence compiler output**

```
policy_I_received_before:
  retrieval_gate_priority: []
policy_I_received_after:
  retrieval_gate_priority: [T5_DFT_VIP]
priority_I_applied: [T5_DFT_VIP]
retrieval_call_I_executed:
  receipt_id: tool_634fe985e864b624
  status: ok
  query: electrolyte solvent anion cluster VIP charge transfer
  top_k: 24

my_updated_evidence_state:
  artifact_id: a_bfed20becf871a8a
  cards: 82
  source_type_counts: {literature_rag: 79, user_evidence: 3}
  queries_added:
  - LiBF4 electrolyte oxidation stability VIP cluster
  - VC additive high-voltage oxidation stability DFT VIP
  - solvent cluster VIP electrolyte oxidation window
  compiler_audit:
    compiler: failed_gate_quota_v2_four_cards_per_gate
    critical_cards: 3
    global_first_n_truncation: false
```

Source: Update/U2_after_R02/04_tool_use.md and 05_evidence_update.md

The R03 prompt carried two active skills, the uncertain and clear R02 failure signatures, and a local archive of 15 entries, of which five entries had current metric payloads in that request. The prompt audit explicitly warned that archive entry count was not independent evaluation count. The proposer therefore received enough information to compare actual outcomes without treating metadata-only entries as extra experiments. Its typed hypotheses proposed salt–solvent rebalancing, alternative salt environments, and compatible solvent motifs, while the numerical composer generated 20 candidates with two exploitation, eleven mixed-salt, three repair, and four diversity-control roles.

**U2 prompt and R03 numerical allocation**

```
my_current_context:
  active_skill_ids:
  - s_t5_r01_ffb9ba
  - s_t5_r02_493c12
  failure_signatures: [two R02 T5_DFT_VIP failures]
  archive_context:
    entries: 15
    entries_with_metrics: 5
    stored_status_counts: {FAILED: 4, PASS: 11}
    entry_count_is_not_independent_evaluation_count: true

my_effective_operator_weights:
  mutate_ratio: 7.0
  seed_from_low_discrepancy_design: 3.0
  probe_failed_gate_boundary: 4.0
  split_primary_salt: 1.0
operators_in_my_generated_batch:
  split_primary_salt: 11
  archive_ratio_or_component_mutation: 5
  seed_from_low_discrepancy_design: 4
my_batch_allocation:
  exploit: 2
  mixed_salt: 11
  repair: 3
  diversity_control: 4
```

Source: Update/U2_after_R02/06_prompt.md and 08_compose_update.md

The important change from U1 to U2 is not simply that more candidates were generated. The workflow learned that different failure semantics require different control actions. Local mutation addressed confidence margin, low-discrepancy seeding addressed coverage, the retrieval priority kept evidence collection focused on the measured gate, and the prompt preserved the distinction between

observed failure and prospective design target. This is a policy update that narrows the next question without altering the physical test.

**Iteration U3, R03 to R04: using a local contrast to choose a parent and a controlled perturbation.** R03 returned two PASS results, two QC-supported VIP misses, and one numerical instability under the same derived $8.2240167\,\mathrm{eV}$ threshold. Candidate `T5_r03_0037_c3eb8d78` had VIP $8.1761454\,\mathrm{eV}$ and pass probability $0.359$. Candidate `T5_r03_0051_9d95d23f` had VIP $7.6303470\,\mathrm{eV}$ and pass probability $0.028$, while its transport screen was $8.2069\,\mathrm{mS\,cm^{-1}}$. Candidate `T5_r03_0003_8470d488` was numerically unstable and had no VIP value. The memory layer retained the same frozen gate with each result and passed only the two measured VIP misses forward as chemical failure signatures.

**U3 task memory: retaining the gate and separating process failure**

```
my_completed_iteration: 3
my_recorded_counts:
  external_submitted: 5
  new_valid: 2
  valid_pool_total: 7
  external_pass: 2
  external_failed: 3

- candidate_id: T5_r03_0037_c3eb8d78
  status: FAILED
  failure_codes: [THRESHOLD_MISS]
  vip_eV: 8.176145391557746
  vip_gate_eV: 8.224016743481572
  vip_pass_probability: 0.359
  NE_mobility_mS_cm: 15.308523367134109

- candidate_id: T5_r03_0051_9d95d23f
  status: FAILED
  failure_codes: [THRESHOLD_MISS]
  vip_eV: 7.630347047244106
  vip_gate_eV: 8.224016743481572
  vip_pass_probability: 0.028
  NE_mobility_mS_cm: 8.206881490514478

- candidate_id: T5_r03_0003_8470d488
  status: FAILED
  failure_codes: [NUMERICAL_INSTABILITY]
  vip_eV: null
  vip_pass_probability: null
```

Source: Update/U3_after_R03/01_task_memory.md

Task memory then made a local comparison that could inform the next search step without being promoted to a causal claim. The failed DMC-containing candidate and the successful archive candidate `T5_r02_0089_7fd0f497` shared $LiPF_6$, SL, BTFE, and FEC inputs. The successful candidate used AN in the corresponding solvent position and had VIP $8.6240261\,\mathrm{eV}$ with pass probability 1.0. The record kept both candidate identities and their separate physical result identifiers. This is exactly the type of counterexample that prevents a global DMC blacklist or a global AN preference. The reflector translated this comparison into the next search intent, adding `mutate_ratio` and `split_primary_salt` while retaining the VIP-focused retrieval priority. Although rollback was eligible and the targeted VIP failure rate rose from 0.2 to 0.4, the valid increment remained two, so the rollback condition was not met and `should_rollback` stayed `false`. The new skill `s_t5_r03_420bd4` was then appended to the first two skills, with weight 2.0 for both ratio mutation and primary-salt splitting.

**U3 local contrast, reflection, and skill update**

```
failed_candidate: T5_r03_0051_9d95d23f
failure_gate: T5_DFT_VIP
failure_vip_eV: 7.630347047244106
successful_counterexample: T5_r02_0089_7fd0f497
```

```
counterexample_vip_eV: 8.624026124096279
shared_inputs: [LiPF6, SL, BTFE, FEC]
local_difference: DMC versus AN

my_submitted_intent:
  focus_gate_ids: [T5_DFT_VIP]
  recommended_operators:
  - mutate_ratio
  - split_primary_salt
rollback_check_received:
  current_valid_increment: 2
  previous_valid_increment: 2
  eligible: true

my_new_skill:
  skill_id: s_t5_r03_420bd4
  failure_refs:
  - history://T5/failure/f_9e41bb4927b1723a
  - history://T5/failure/f_96dca78ae2a84a60
  machine_prior:
    operator_weights:
      mutate_ratio: 2.0
      split_primary_salt: 2.0
    evidence_focus_gates: [T5_DFT_VIP]
    failed_cases_in_next_request: 2
```

Source: case study/case.md and Update/U3_after_R03/02_reflection.md and 03_skill.md

The tool use and evidence artifacts preserved the existing priority but changed the query content. The retriever searched for $LiPF_6$ solvation, anion–solvent cluster ionization, and the DMC/AN environment rather than changing source weights or treating the earlier charge-transfer card as direct candidate evidence. The evidence bundle for R04 contained 92 cards, 89 literature cards and three user or task cards. The selected contextual card still had no extracted protocol and no assessed uncertainty. The prompt carried three active skills, two R03 VIP failures, and an archive with 15 entries and seven current metric payloads. The model role instruction remained unchanged.

**U3 tool, evidence, and prompt transition**

```
retrieval_gate_priority_before: [T5_DFT_VIP]
retrieval_gate_priority_after: [T5_DFT_VIP]
queries_added:
- \(\Delta\)SCF VIP anion-solvent cluster electrolyte
- LiPF6 solvation cluster ionization DFT electrolyte
- solvent environment cluster VIP DMC AN

my_updated_evidence_state:
  artifact_id: a_633f97efbfda0bed
  cards: 92
  source_type_counts: {literature_rag: 89, user_evidence: 3}
  selected_contextual_card:
    evidence_id: ev_cee27af3f3b35a19fc6f
    direct_to_failed_gate: false
    protocol: not_extracted
    uncertainty: not_assessed
  tool_policy_gate_priority_applied: [T5_DFT_VIP]

my_current_context:
  active_skill_ids:
  - s_t5_r01_ffb9ba
  - s_t5_r02_493c12
  - s_t5_r03_420bd4
  failure_signatures: [two R03 T5_DFT_VIP failures]
  archive_context:
    entries: 15
    entries_with_metrics: 7
    stored_status_counts: {FAILED: 6, PASS: 9}
    entry_count_is_not_independent_evaluation_count: true
```

Source: Update/U3_after_R03/04_tool_use.md, 05_evidence_update.md, and 06_prompt.md

The proposer returned three typed directions. The direct hypothesis targeted the failed DMC environment and recommended component swapping, ratio mutation, and failed-gate boundary probing. An adjacent hypothesis tested mixed-anion balance through primary-salt splitting. A speculative hypothesis recombined compatible successful motifs. All three hypotheses named the gate, component scope, mechanism, operator list, evidence references, and confidence class. None emitted concentrations, predicted VIP values, thresholds, or PASS labels.

**U3 typed hypothesis interface**

```
hypotheses_I_submit:
- hypothesis_id: hyp_7c76b5fffa21f860
  target_gate_ids: [T5_DFT_VIP]
  component_ids: [LiPF6, SL, DMC, BTFE, FEC, AN]
  mechanism: oxidation_resistance
  recommended_operators:
  - swap_component
  - mutate_ratio
  - probe_failed_gate_boundary
  confidence_class: direct
  evidence_refs:
  - ev_eb552160d08b73d3dd7d
  - ev_44b67b490e116fb789c4
  - ev_feed84cd6d766508a5a7
  - ev_2b8189adcbe56aac87d1
  - ev_cee27af3f3b35a19fc6f
numeric_composition_is_my_output: false
```

Source: Update/U3_after_R03/07_hypothesis_update.md

The composer used the successful AN-containing candidate as a parent rather than mutating the failed DMC candidate blindly. It changed the relative SL input from 0.189245 to 0.207518 for candidate `T5_r04_0016_6f5f0f4e` and the relative AN input from 0.203394 to 0.199176 for candidate `T5_r04_0028_763f530f`. Both children retained the parent identity and the same hypothesis ID. The composition record also states that the stored solvent quantities are relative inputs and that the numeric values were not generated by the language model. The normalized fractions reported below are derived for readability and do not replace the recorded inputs used by the composer.

**U3 composer: parent linkage and numerical child generation**

```
skills_I_received:
- s_t5_r01_ffb9ba
- s_t5_r02_493c12
- s_t5_r03_420bd4
operators_in_my_generated_batch:
  archive_ratio_or_component_mutation: 6
  split_primary_salt: 9
  seed_from_low_discrepancy_design: 5

- candidate_id: T5_r04_0016_6f5f0f4e
  parent_ids: [T5_r02_0089_7fd0f497]
  hypothesis_id: hyp_7c76b5fffa21f860
  operator: archive_ratio_or_component_mutation
  changed_input: SL 0.18924530287093141 -> 0.20751822516515775
  llm_generated_numeric_values: false

- candidate_id: T5_r04_0028_763f530f
  parent_ids: [T5_r02_0089_7fd0f497]
  hypothesis_id: hyp_7c76b5fffa21f860
  operator: archive_ratio_or_component_mutation
  changed_input: AN 0.20339449657527903 -> 0.1991758052456049
  llm_generated_numeric_values: false
```

Source: Update/U3_after_R03/08_compose_update.md

External validation then returned independent results for the two new candidate IDs. Candidate `T5_r04_0016_6f5f0f4e` reached VIP $8.8647749\,\mathrm{eV}$, pass probability 1.0, and transport

19.5596 mS cm$^{-1}$. Candidate `T5_r04_0028_763f530f` reached VIP 8.8863042 eV, pass probability 1.0, and transport 13.7259 mS cm$^{-1}$. Both had QC-supported PASS results. The evidence supports the usefulness of probing this local neighborhood, but it does not show that AN alone caused the improvement. The causal limitation is preserved because the two descendants, their parent, their operator, and their physical results remain separately linked.

**The concrete R04 formulation obtained after the U3 update**

```
candidate_id: T5_r04_0016_6f5f0f4e
parent_id: T5_r02_0089_7fd0f497
hypothesis_id: hyp_7c76b5fffa21f860
components:
- component_id: LiPF6
  value: 1.7013827689501908
  unit: mol_L
- component_id: SL
  value: 0.20751822516515775
  unit: solvent_ratio
- component_id: AN
  value: 0.20339449657527903
  unit: solvent_ratio
- component_id: BTFE
  value: 0.4515101633744687
  unit: solvent_ratio
- component_id: FEC
  value: 0.15585003717932086
  unit: solvent_ratio
normalized_molar_fractions:
  SL: 0.203794
  AN: 0.199745
  BTFE: 0.443408
  FEC: 0.153053
normalization_sum_of_relative_inputs: 1.018273
normalization_for_reporting_only: true
operator: archive_ratio_or_component_mutation
llm_generated_numeric_values: false
vip_eV: 8.864774912408254
vip_pass_probability: 1.0
NE_mobility_mS_cm: 19.55963288215083
decision: PASS
```

Source: Update/U3_after_R03/08_compose_update.md and R04 external evaluator record

**Iteration U4, R04 to R05: consolidating the successful region and retaining the unresolved boundary.** R04 produced four PASS results and one QC-supported VIP threshold miss. The remaining failure was candidate `T5_r04_0034_316b0938`, a mixed LiTFSI/$LiPF_6$ formulation containing BTFE, AN, FEC, SL, and GBL. Its VIP was 7.4637983 eV against the derived 8.2240167 eV threshold, its pass probability was 0.005, and its transport result was 11.4583 mS cm$^{-1}$. Task memory kept this failure active even though the batch was mostly successful. It did not attribute the miss to LiTFSI alone because the salt proportions and several solvent quantities also differed from those of the successful candidates.

**U4 task memory and persistent failure boundary**

```
my_completed_iteration: 4
my_recorded_counts:
  external_submitted: 5
  new_valid: 4
  valid_pool_total: 11
  external_pass: 4
  external_failed: 1

remaining_failure:
  candidate_id: T5_r04_0034_316b0938
  result_id: mdr_a6ddb00b358207e32647
  failure_codes: [THRESHOLD_MISS]
  vip_eV: 7.4637983493514195
  vip_ci90_eV: [7.236763816901347, 7.586046136151053]
```

```
  vip_gate_eV: 8.224016743481572
  vip_pass_probability: 0.005
  NE_mobility_mS_cm: 11.458310038205747

successful_neighbors_retained: 4
```

Source: Update/U4_after_R04/01_task_memory.md

The reflector changed the search objective from finding a feasible region to preserving a measured useful neighborhood while probing the remaining boundary. It recommended `probe_failed_gate_boundary` and `seed_from_evidence`. The rollback check was eligible, but `should_rollback: false` because the current valid increment was four, the previous increment was two, and the targeted failure-rate condition did not request removal. The fourth skill, `s_t5_r04_d5be23`, was appended to the previous three skills, with the remaining failure reference and five pass counterexamples retained in the next request.

**U4 reflection and cumulative skill ledger**

```
my_submitted_intent:
  focus_gate_ids: [T5_DFT_VIP]
  recommended_operators:
  - probe_failed_gate_boundary
  - seed_from_evidence
rollback_check_received:
  current_valid_increment: 4
  previous_valid_increment: 2
  eligible: true

my_active_skill_ids_after:
- s_t5_r01_ffb9ba
- s_t5_r02_493c12
- s_t5_r03_420bd4
- s_t5_r04_d5be23
my_new_skill:
  skill_id: s_t5_r04_d5be23
  failure_refs: [history://T5/failure/f_be727dce30a76db7]
  machine_prior:
    operator_weights:
      probe_failed_gate_boundary: 2.0
      seed_from_evidence: 2.0
    evidence_focus_gates: [T5_DFT_VIP]
    failed_cases_in_next_request: 1
    pass_counterexamples_in_next_request: 5
```

Source: Update/U4_after_R04/02_reflection.md and 03_skill.md

Tool use retained the persistent `T5_DFT_VIP` priority but made the query content more specific to the unresolved environment. The retriever queried ΔSCF anion–solvent cluster ionization, LiTFSI/$LiPF_6$ acetonitrile–sulfolane environments, and high-voltage solvent-cluster ionization. The evidence bundle contained 86 cards, 83 literature cards and three user or task cards. Its compiler audit retained the three critical evidence references, applied the existing source weights, and recorded that the gate priority had actually been applied. This is a change in information routing rather than a claim that the literature directly validated the failing formulation.

**U4 tool use and evidence update**

```
policy_I_received_before:
  retrieval_gate_priority: [T5_DFT_VIP]
priority_I_applied: [T5_DFT_VIP]
retrieval_call_I_executed:
  receipt_id: tool_1eeca890dc9980f9
  status: ok
  task_id: T5
  top_k: 24
```

```
queries_added:
- \(\Delta\)SCF VIP anion-solvent cluster electrolyte
- LiTFSI LiPF6 acetonitrile sulfolane cluster oxidation potential DFT
- high-voltage electrolyte solvent cluster ionization energy DFT

my_updated_evidence_state:
  artifact_id: a_dbcd817d0498e87c
  cards: 86
  source_type_counts: {literature_rag: 83, user_evidence: 3}
  current_failure_signatures: [one R04 T5_DFT_VIP failure]
  tool_policy_gate_priority_applied: [T5_DFT_VIP]
```

Source: Update/U4_after_R04/04_tool_use.md and 05_evidence_update.md

The R05 prompt carried four active skills, the remaining R04 failure, and the successful local archive. Its context audit reported 15 archive entries and nine entries with current metrics. Separately, the cumulative external archive contained 11 PASS and 9 FAILED candidates after four five-candidate evaluations. These counts belong to the illustrative four-window excerpt and are not the full 80-candidate task 5 aggregate reported in the main results. The fixed proposer instruction still prohibited numerical composition and physical verdicts. This matters because merely receiving four skills does not imply that a valid new hypothesis was produced.

**U4 prompt context at the R05 boundary**

```
my_current_context:
  active_skill_ids:
  - s_t5_r01_ffb9ba
  - s_t5_r02_493c12
  - s_t5_r03_420bd4
  - s_t5_r04_d5be23
  failure_signatures:
  - signature_id: fs_18bac054c9c4ca22
    failed_gate_id: T5_DFT_VIP
    failure_refs: [history://T5/failure/f_be727dce30a76db7]
  archive_context:
    entries: 15
    entries_with_metrics: 9
    stored_status_counts: {FAILED: 4, PASS: 11}
    entry_count_is_not_independent_evaluation_count: true
  context_audit:
    hard_signals_dropped: false
```

Source: Update/U4_after_R04/06_prompt.md

At the R05 hypothesis interface, the model returned an empty final response. The runtime did not promote hidden reasoning text to a typed hypothesis and did not invent a missing JSON object. It recorded `final_model_hypothesis_available: false` and passed a deterministic evidence-conditioned fallback with `seed_from_low_discrepancy_design` and `split_primary_salt`. The fallback retained the unchanged task-level gate context and the evidence references. It was labeled speculative rather than model-authored.

**U4 typed fallback after an unusable model response**

```
final_model_hypothesis_available: false
fallback_I_pass_to_composition:
- hypothesis_id: hyp_bf7d3864bd81b377
  component_ids: []
  interaction_scope: {}
  mechanism: other
  recommended_operators:
  - seed_from_low_discrepancy_design
  - split_primary_salt
  evidence_refs:
  - ev_44b67b490e116fb789c4
  - ev_eb552160d08b73d3dd7d
```

```
  - ev_feed84cd6d766508a5a7
  confidence_class: speculative
  rationale: deterministic evidence-conditioned DoE fallback after no valid typed
    model hypothesis
numeric_composition_is_my_output: false
```

Source: Update/U4_after_R04/07_hypothesis_update.md

The composer continued with `ARCHIVE_GUIDED_NSGA2` and generated 20 candidates using four exploitation, eight mixed-salt, four repair, and four diversity-control roles. The effective operator weights were 6.0 for ratio mutation, 4.0 for failed-gate boundary probing, 3.0 for low-discrepancy seeding, 3.0 for primary-salt splitting, and 2.0 for evidence seeding. The batch record preserved the fallback hypothesis ID and recorded zero model-hypothesis component coverage. That explicit zero is important because it prevents a downstream report from claiming that the batch was guided by a completed model hypothesis.

**U4 composer continuation after fallback**

```
skills_I_received:
- s_t5_r01_ffb9ba
- s_t5_r02_493c12
- s_t5_r03_420bd4
- s_t5_r04_d5be23
my_effective_operator_weights:
  seed_from_low_discrepancy_design: 3.0
  split_primary_salt: 3.0
  mutate_ratio: 6.0
  probe_failed_gate_boundary: 4.0
  seed_from_evidence: 2.0
operators_in_my_generated_batch:
  archive_ratio_or_component_mutation: 8
  split_primary_salt: 8
  seed_from_low_discrepancy_design: 4
my_batch_allocation:
  exploit: 4
  mixed_salt: 8
  repair: 4
  diversity_control: 4
llm_generated_numeric_values: false
model_hypothesis_component_coverage: 0
```

Source: Update/U4_after_R04/08_compose_update.md

The four windows show a progressively more constrained search question. U1 asked which local changes might repair two clear VIP misses. U2 asked how to obtain a reliable margin around an uncertain point while escaping one unsuccessful neighborhood. U3 asked whether the solvent environment and salt balance could explain a local DMC-versus-AN contrast, while explicitly retaining the causal limitation. U4 asked how to preserve a measured successful neighborhood while mapping the boundary of a remaining mixed-salt failure. These questions became more specific because each was anchored to candidate identifiers, measured values, the fixed gate identifier, and source-linked evidence. The skill ledger shows cumulative rather than overwrite semantics. U1 registered `s_t5_r01_ffb9ba`, U2 appended `s_t5_r02_493c12`, U3 appended `s_t5_r03_420bd4`, and U4 appended `s_t5_r04_d5be23`. Together they formed a task-specific design memory grounded in linked failures and evidence, rather than a sequence of untraceable prompt rewrites. The prompt ledger shows a second form of iteration. R01 began with no active skills, no measured local archive, and only prospective gate signatures. R02 added one skill, two observed VIP failures, and five metric-bearing results. R03 added a second skill together with the R02 uncertainty and threshold-miss records. R04 added a third skill and the measured DMC/AN contrast. R05 carried four skills, one remaining failure, and nine metric-bearing archive entries. The fixed role instruction did not change. Only the typed context changed. The tool and evidence ledger further shows why retrieval is part of self-improvement. U2 committed `retrieval_gate_priority=[T5_DFT_VIP]`, and U3–U4 retained that priority while shifting queries to the relevant solvent and mixed-salt environments. Evidence-card counts changed from 84 to 82 to 92 to 86, but the key update was not

the raw count. It was the link between the current failure signature, retrieved evidence, the query receipt, and the next typed hypothesis. Finally, the composition ledger separates qualitative intent from numerical execution. It records how weighted operators became concrete candidate batches, how some allocations were introduced by runtime configuration, how selected children were tied to successful parents and named hypotheses, and how fallback batches preserved the same search mode after a model failure. Together, these records show how a hypothesis becomes a formulation through typed interaction scope, deterministic parent or seed selection, numerical candidate generation, and independent physical evaluation.

Across the illustrative R01–R04 excerpt, the valid pool increased from 3 to 11, and the external outcomes were 3/5, 2/5, 2/5, and 4/5 passes. The improvement is not monotonic and should not be interpreted as proof that any solvent or salt is universally optimal. Instead, this case illustrates a controlled transformation of physical failure into search state. Candidate-level memory preserves observations, reflection assigns failure-specific actions, skill stores them for reuse, retrieval and evidence updates focus information gathering, prompt context carries the relevant state, typed hypotheses define testable interactions, and the numerical composer instantiates candidates under frozen constraints. The R04 descendants show the concrete outcome of this process. Both descendants of the successful AN-containing parent passed independent external evaluation, with VIP values of $8.8648$ and $8.8863\,\mathrm{eV}$, pass probability 1.0, and transport values of $19.5596$ and $13.7259\,\mathrm{mS\,cm^{-1}}$. These records support a useful local search direction, not a universal AN rule or a causal decomposition of mixture effects. They also do not establish a five-volt electrochemical window, CEI stability, long-term cycling, gas suppression, or experimental cell performance. Those claims require additional controlled simulations or experiments. What the T5 case demonstrates is that the framework can preserve failures, counterexamples, uncertainty, evidence provenance, and numerical authority while using each grounded update to make the next search more specific, more auditable, and more robust to an unusable model response.

### A.11 Limitations and Future Work

The primary limitation of this work is the lack of wet-lab experimental validation. Although SAGE is evaluated using a consistent computational pipeline and physically grounded criteria, computational assessment alone cannot fully establish whether the proposed electrolyte formulations will realize the same performance in real experimental settings. In particular, properties that appear favorable *in silico* are based partly on computational screening proxies and may still be affected by synthesis feasibility, material purity, interfacial side reactions, measurement variability, and other laboratory-specific factors that are not fully captured by the current evaluation framework. Accordingly, an important direction for future work is to close the loop between computational design and experimental verification. This includes validating the most promising candidates through wet-lab testing, comparing predicted and measured performance under practical operating conditions, and using experimental results to further refine the search process. More broadly, future work may explore tighter integration with automated experimentation platforms and iterative design–build–test cycles, enabling SAGE to serve not only as a computational discovery system but also as a more complete framework for real-world scientific optimization.