SCOPE: SIBLING-CONTRAST PROCEDURAL EVOLUTION FOR SELF-EVOLVING POWER GRID PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Model (LLM) agents can use external simulators to verify proposed actions in power grid planning, but turning the resulting search experience into reusable planning knowledge remains a challenge. We introduce Sibling-Contrast Procedural Evolution (SCOPE), a self-evolving framework for secure power grid reinforcement planning that learns reusable planning strategies from verified search experience. Within each episode, SCOPE generates complementary candidate plans guided by diagnostic evidence, investment cost, and security requirements, and evaluates their physical consequences from the same grid state. It then performs plan fusion by selecting and recombining actions from the candidates before selecting an improved plan. Across episodes, SCOPE maintains two external memory banks: an episodic bank that stores concrete precedents from previously solved scenarios, and a procedural bank that consolidates verified planning trajectories into reusable planning strategies that guide future tasks. We evaluate SCOPE on 100 operating scenarios of an augmented IEEE 118-bus system using 24-hour AC power flow checks and peak hour N-1 contingency analysis. Results show that SCOPE can continuously improve its power planning strategies across episodes and provide better plans for future scenarios.

1 INTRODUCTION

Large language model (LLM) agents can combine language-based reasoning with external simulators to address complex planning tasks (Jin et al., 2025; Zhang et al., 2025c). Power grid reinforcement planning presents two connected challenges for such agents: selecting effective interventions within a scenario and reusing experience across scenarios. A planner must choose equipment upgrades, new lines, and reactive support that resolve security violations while controlling investment cost. Each proposal requires physical evaluation because its consequences depend on the network and operating conditions. Effective planning requires both searching complementary interventions and learning from resulting experience for subsequent decisions.

These planning and learning challenges are closely connected. Within an episode, a line upgrade may reduce overloads while reactive support improves voltage conditions. Comparing such alternatives from the same retained state helps identify complementary actions, whose combination requires separate verification because their effects can interact. The resulting experience can also guide later scenarios: selected interventions reveal effective responses, while unselected alternatives expose partial improvements, unnecessary investment, or adverse effects. Together, these observations can support conditional procedures, such as targeting remaining contingency bottlenecks after resolving base-case violations. Reusing this knowledge requires preserving useful planning principles while adapting specific actions to the current equipment and operating conditions.

Prior work has developed reasoning-and-action frameworks, search-based agents, and mechanisms for learning from experience (Yao et al., 2022; Zhou et al.; Zhao et al., 2024; Ouyang et al.). Adapting these methods to power grid planning requires addressing two challenges. First, reinforcement decisions must balance multiple coupled security constraints together with investment cost: an intervention that reduces overloads may leave voltage violations unresolved, and combining useful actions from different plans can alter their physical effects. General candidate exploration and ranking therefore require explicit mechanisms for comparing security outcomes and verifying action combinations. Second, past solutions and extracted strategies depend on the operating conditions

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.

and equipment involved. Their reuse requires identifying when the underlying guidance applies and adapting it to current physical diagnostics. These challenges motivate SCOPE's integration of complementary sibling search, constrained and separately verified plan fusion, and conditional strategy learning from verified experience.

We introduce Sibling-Contrast Procedural Evolution (**SCOPE**). Within each episode, SCOPE uses *verifier-grounded sibling search and plan fusion*: it generates three complementary candidates guided by diagnostic evidence, cost, and security constraints, evaluates them from a shared network state, synthesizes a non-conflicting fused alternative, and selects the best intervention after physical verification. Across episodes, SCOPE consolidates verified search outcomes into two training-free external memories: episodic memory supplies concrete past precedents at episode onset, while procedural memory provides conditional strategies once planning feedback arrives. Search supplies verified data to memory, and retrieved experience refines subsequent sibling proposals—ensuring every candidate remains strictly physically verified.

We evaluate SCOPE on 100 scenarios across five operating conditions on an augmented IEEE 118-bus system using 24-hour AC power-flow and peak-hour $N-1$ checks. Component comparisons assess sibling search, plan fusion, and memory reuse. With memories frozen during held-out evaluation, SCOPE achieves 100% secure completion and, relative to the same search without memory, halves median planning rounds and reduces median completion time and output tokens by 52.3% and 58.4%, respectively, with similar median reinforcement cost.

Our contributions are threefold: ❶ **Verifier-grounded sibling search and plan fusion.** We develop a reinforcement planning mechanism that generates complementary candidates, evaluates them from a common retained state, and constructs a constrained fusion of their proposed actions for separate verification before commitment. ❷ **Learning and reusing planning experience.** We develop complementary episodic and procedural memories that preserve concrete precedents and extract conditional strategies from selected and informative unselected outcomes, guiding subsequent planning with fixed LLM parameters. ❸ **Evaluation of planning and experience reuse.** We quantify the contributions of search, fusion, and memory through component comparisons on held-out operating scenarios, together with cost refinement and analyses of successful and unsuccessful strategy reuse.

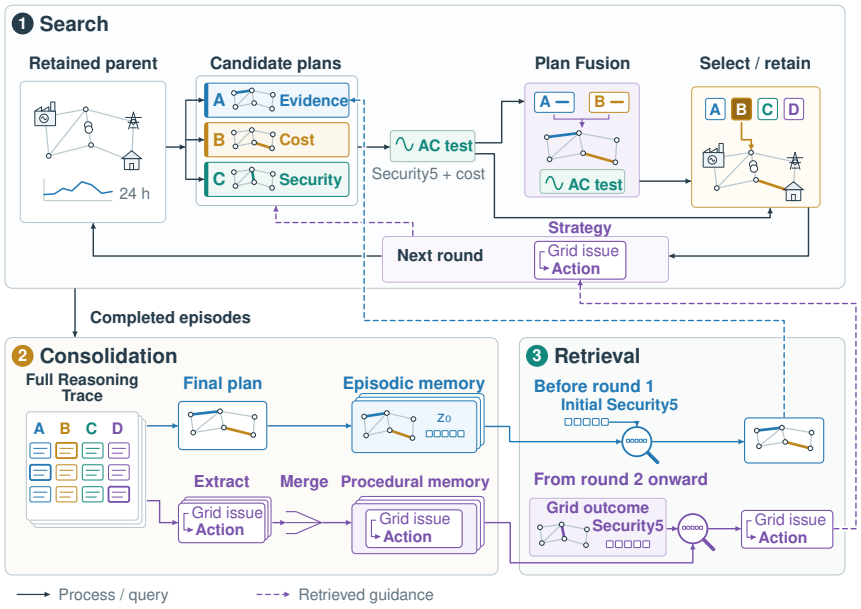


Figure 1: **Overview of SCOPE framework.** Within an episode, SCOPE evaluates sibling reinforcement plans from a shared retained state, verifies a fused alternative, and commits an improved plan. Across episodes, verified planning experience is retained and consolidated into reusable guidance for future planning.

2 Related Work

Research on self-evolving LLM agents studies how interaction experience can improve skills, workflows, contexts and policies. (Wang et al., 2023) accumulates executable skills through environmental interaction, while (Zhang et al., 2025a) optimizes agentic workflows through Monte Carlo tree search and execution feedback. (Zhang et al., 2025b) maintains evolving contextual playbooks through generation, reflection, and curation. (Wu et al.) combines the distillation of reusable strategic principles with reinforcement learning updates to the agent policy. These approaches differ in what changes as experience accumulates.SCOPE evolves through accumulated planning experience while keeping the LLM parameters fixed. Verified reinforcement searches provide the evidence from which it builds and refines strategies for future scenarios.

Agent memory supports the reuse of experiences and the extraction of reusable procedures. (Shinn et al., 2023) stores verbal reflections on task feedback, while (Zhao et al., 2024) combines retrieved experiences with insights extracted from both successful and unsuccessful attempts. (Wang et al., 2024) induces reusable workflows and (Fang et al., 2026) studies the construction, retrieval and updating of procedural memory. (Zhang et al., 2026) learns utility estimates for episodic memories from environmental feedback to guide retrieval. Closely related, (Ouyang et al.) distills strategies from successful and failed trajectories, its memory aware test time scaling framework compares multiple trajectories to improve memory construction. SCOPE learns from the verified outcomes of both accepted and rejected reinforcement alternatives evaluated from the same parent plan. Episodic memory preserves concrete precedents, while procedural memory extracts reusable planning strategies. Whereas ReasoningBank uses an LLM-as-a-judge to assess outcomes, SCOPE grounds security assessments in AC power flow and $N-1$ checks.

LLM agents also use environmental feedback to guide planning and action selection. (Yao et al., 2022) interleaves reasoning and actions with respect to environmental observations, while (Zhou et al.) uses search, value estimation and reflection to explore alternative trajectories. In power systems, (Jin et al., 2025) coordinates AC optimal power flow and contingency analysis tools, and (Zhang et al., 2025c) validates corrective actions through sanboxed execution and rollback. (Chen et al., 2025) combines hierarchal tool orchestration with retrieval and prompt refinement using human feedback. These systems establish precedents for both numerical verification and adaptation in grid agents. SCOPE connects reinforcement search with learning across scenarios. It compares three complementary candidates from the same parent plan, independently verifies combinations of their tested actions through constrained plan fusion, and turns the experience into strategies that guide future searches.

3 Problem Formulation

3.1 Planning Setting

We consider a transmission network with initial topology $\mathcal{G}^0 = (\mathcal{B}, \mathcal{E}^0)$, where $\mathcal{B}$ contains the buses and $\mathcal{E}^0$ contains the existing lines and transformers. Each operating scenario $s \in \mathcal{S}$ specifies 24 hourly profiles of active demand, renewable availability, and conventional generator availability. Reactive demand preserves each load's baseline power factor, and renewable injections supply their available active power at zero reactive power. These scenario profiles remain fixed throughout an episode, while the planner modifies equipment ratings, network topology, and reactive support through a reinforcement plan. The network and scenario data are described in Appendix A

Reinforcement actions and plans. The planner uses four operation types, $\mathcal{A} = \{\text{line upgrade}, \text{transformer upgrade}, \text{new-line construction}, \text{reactive support}\}$. An instantiated action $u = (a, \xi, v)$ specifies an operation $a \in \mathcal{A}$, the affected asset or installation location ξ, and its final target value v. Let $\mathcal{U}$ denote the set of admissible instantiated actions. A reinforcement plan $\pi \in \Pi$ is a finite collection of compatible actions from $\mathcal{U}$, with no conflicting final targets assigned to the same asset or installation location. Applying π updates the simulator model according to these targets, including equipment ratings and their associated electrical parameters, planned-line parameters, and bus-level reactive support. The exact action fields, target domains, and simulator parameter mappings are given in Appendix A.3.

Investment cost. The investment cost $C(\pi)$ of a plan is the sum of the costs of its actions, measured relative to the original network. It counts only the final reinforcement i.e candidates tested and rejected during planning are not charged, and a revised target is charged once, at its final value. Each action costs a fixed amount plus a term proportional to its size, namely the increase over the original rating for an upgrade, the rating of a new line, or the magnitude of reactive support. The exact cost functions are given in Appendix A.3.

3.2 Physical Verification and Security State

The LLM does not decide whether a plan is secure; an external verifier does. The verifier applies π to the simulator model and evaluates the resulting network using AC power flow (`pandapower`) (Thurner et al., 2018). It evaluates all 24 base-case operating hours and performs $N-1$ contingency analysis at the peak-load hour. Each evaluation returns bus voltages and branch loadings, which are checked against the security limits below. The AC power-flow equations are given in Appendix A.4.

Security constraints. The base case covers all 24 hours. In every hour, every bus voltage must lie within $[0.95, 1.05]$ p.u., and no in-service line or transformer may be loaded above 100%. The $N-1$ check is performed at the hour of maximum total load by removing each line and transformer, including new lines, one at a time. Because grid operators typically permit relaxed short-term emergency limits following a contingency, the $N-1$ screen uses a 110% branch-loading limit and a 0.92 p.u. minimum-voltage limit. A contingency is unacceptable if either limit is violated.

Security5. A binary secure/insecure outcome does not indicate the type or magnitude of remaining violations. We therefore summarize the verified outcome of plan π under scenario s as

$$\mathbf{z}(\pi, s) = [T_B, V_B, T_{N1}, V_{N1}, F_{N1}], \tag{1}$$

which we call *Security5*. In order, its components are (i) the *base-case thermal severity* T_B, which measures how far branch loadings exceed the 100% limit over the 24 hours, (ii) the *base-case voltage severity* V_B, which measures how far bus voltages fall below 0.95 p.u. or rise above 1.05 p.u., (iii) the $N-1$ *thermal severity* T_{N1}, which measures how far surviving branches exceed the 110% emergency limit under the peak-hour contingencies, (iv) the $N-1$ *voltage severity* V_{N1}, which measures how far bus voltages fall below 0.92 p.u. under these contingencies and (v) the $N-1$ *failure fraction* F_{N1}, the fraction of tested contingencies that are unacceptable. Each severity is the root mean square (RMS) of the violations relative to the limit over all branches or buses, where those within limits count as zero. For example, a branch loaded at 105% in the base case contributes 0.05 to T_B. Lower values are better in all five components.

Security5 is recomputed for every candidate plan, so new lines are included in both the base-case and $N-1$ evaluations. A plan is therefore secure only if $\mathbf{z}(\pi, s) = \mathbf{0}$. Along with Security5, the verifier returns diagnostics listing the remaining violations, which are given to the planner (details in Appendix A.5).

3.3 Planning Objective

For each scenario s, the goal is to find the cheapest reinforcement plan that makes the network secure:

$$\pi^\star(s) \in \arg\min_{\pi \in \Pi} C(\pi) \quad \text{subject to} \quad \mathbf{z}(\pi, s) = \mathbf{0}. \tag{2}$$

SCOPE uses a heuristic search for Eq. (2) and does not guarantee a globally optimal plan. During the main search, a candidate is accepted if it improves the Security5 vector componentwise or preserves the same Security5 vector at a lower investment cost. The main search terminates when the retained plan satisfies the security criterion or the planning budget is exhausted. Once a secure plan is obtained, additional refinement rounds may be used to further reduce its investment cost while preserving security, this cost reduction is treated as a secondary objective.

Experience-reuse objective. We evaluate whether experience from other operating scenarios helps the planner reach secure solutions with lower investment cost or reduced planning effort. For each evaluation scenario, memory-based and memory-free configurations use the same network, action space, and security criteria. Their performance is compared through secure completion, investment cost, and computational effort.

Algorithm 1 SCOPE planning for one scenario with fixed memories

Require: scenario s_e, memories $\mathcal{M}_{\text{epi}}$ and $\mathcal{M}_{\text{proc}}$, round budget $k_{\max}$
Ensure: final plan π_e^{fin}
Start from $\pi_{e,0} = \varnothing$ and verify it to obtain $S_{e,0}$
Retrieve the closest episodic entry for Candidate A, if one exists
for $k = 0, 1, \ldots, k_{\max} - 1$ **do**
  if $\pi_{e,k}$ is secure **then break**
  if $k > 0$ **then** retrieve up to two strategies from $\mathcal{M}_{\text{proc}}$
  Generate $\Delta_{e,k}^{\text{A}}$, $\Delta_{e,k}^{\text{B}}$, $\Delta_{e,k}^{\text{C}}$ in one LLM call
  Verify each sibling plan and obtain $o_{e,k}^{c}$ for $c \in \{\text{A}, \text{B}, \text{C}\}$
  If a valid fusion exists, build $\Delta_{e,k}^{\text{D}}$ by Eq. (4) and verify it
  Select $c_{e,k}^{\star}$ by Eq. (5), or $c_{e,k}^{\star} = \varnothing$ if no candidate is eligible
  Record $\mathcal{R}_{e,k}$ by Eq. (3) and update the retained plan to $\pi_{e,k+1}$
end for
return π_e^{fin}, the last retained plan

4 SCOPE: Sibling-Contrast Procedural Evolution

4.1 Framework Overview

SCOPE works in three phases (Figure 1). In the *Search* phase, every planning round generates several candidate plans and records their verified outcomes (Section 4.2). In the *Consolidation* phase, the outcomes of completed episodes, including those of rejected candidates, are converted into reusable strategies (Section 4.3). In the *Retrieval* phase, this memory helps guide the search in later episodes (Section 4.4). The planner LLM is never retrained; only its external memory changes.

Planning rounds and records. Episode e addresses scenario s_e. At round k, its retained state is $S_{e,k} = (\pi_{e,k}, z_{e,k}, d_{e,k})$, where $d_{e,k}$ contains the verifier diagnostics. A candidate update $\Delta_{e,k}^{c}$ is a compatible collection of instantiated actions. It overwrites targets on the assets it modifies and preserves all other retained actions. Admissible updates must produce a plan in Π. Testing a candidate does not change the retained state. Let $\mathcal{K}_{e,k}$ contain the candidates actually evaluated in round k. Their verified outcomes are recorded as $o_{e,k}^{c} = \big(z_{e,k}^{c}, d_{e,k}^{c}, C(\pi_{e,k}^{c})\big)$. The complete round record is

$$\mathcal{R}_{e,k} = \Big(S_{e,k}, \big\{(\Delta_{e,k}^{c}, o_{e,k}^{c})\big\}_{c \in \mathcal{K}_{e,k}}, c_{e,k}^{\star}\Big), \tag{3}$$

where $c_{e,k}^{\star}$ identifies the committed candidate and equals $\varnothing$ when no candidate is committed. The next retained plan is $\pi_{e,k+1} = \begin{cases} \pi_{e,k}^{c_{e,k}^{\star}}, & c_{e,k}^{\star} \neq \varnothing, \\ \pi_{e,k}, & c_{e,k}^{\star} = \varnothing. \end{cases}$. These records provide the evidence used for memory construction. In the experiments, logging evaluation episodes does not update the frozen memory banks.

4.2 Verifier-Grounded Sibling Search and Plan Fusion

Candidate generation. In each round, the planner LLM proposes three updates, Δ_k^{A}, Δ_k^{B}, and Δ_k^{C}, for the current retained plan in a single call, so the three siblings are generated together rather than independently. Each follows a fixed role prompt (Appendix D). Candidate A (*evidence-oriented*) adapts the retrieved past episode to the current diagnostics or when episodic memory is available, the retrieved entry is included in the joint generation prompt with instructions that Candidate A use it to construct an experience-informed proposal, otherwise Candidate A relies on the current diagnostics alone. Candidate B (*cost-oriented*) looks for a focused, low-cost intervention. Candidate C (*security-oriented*) considers broader and more aggressive reinforcement. Only C may build a new line, because a new line changes the topology and has by far the largest fixed cost. A and B therefore work with cheaper upgrades. The three plans must differ, and all start from the same retained plan, so their verified outcomes can be compared directly. The costs of all actions are listed in Appendix A.3.

Plan Fusion. After verifying A, B, and C, the planner LLM constructs a fused update from their proposed actions and verified outcomes:

$$\Delta^{\mathrm{D}}_{e,k} = \mathrm{Fuse}_{\mathrm{LLM}}\left(\left\{(\Delta^{c}_{e,k}, o^{c}_{e,k})\right\}_{c\in\{\mathrm{A,B,C}\}}\right). \tag{4}$$

A valid fusion may select only actions proposed by A, B, or C, preserves their target values, assigns at most one target per asset, and includes actions originating from at least two distinct siblings. Source references retain the origin of each selected action even when identical actions appear in multiple siblings. Candidate D is then independently verified from the same retained plan as A, B, and C. The siblings need not be secure before fusion; their actions may address different remaining violations, while their combination can produce a different physical outcome and therefore requires separate verification.

Selection and commitment. SCOPE compares candidate plans using componentwise Security5 dominance. We write

$$\mathbf{z}_1 \prec_S \mathbf{z}_2 \iff \mathbf{z}_1 \le \mathbf{z}_2 \quad \wedge \quad \mathbf{z}_1 \neq \mathbf{z}_2,$$

so that $\mathbf{z}_1 \prec_S \mathbf{z}_2$ means that the first state is no worse in any Security5 component and strictly better in at least one. For two plans evaluated under the same scenario s, SCOPE prefers π_1 to π_2 when

$$\pi_1 \prec_s \pi_2 \iff \mathbf{z}(\pi_1, s) \prec_S \mathbf{z}(\pi_2, s) \vee \left[\mathbf{z}(\pi_1, s) = \mathbf{z}(\pi_2, s) \wedge C(\pi_1) < C(\pi_2)\right]. \tag{5}$$

A candidate is eligible if it is preferred to the retained plan. Among eligible candidates, SCOPE removes those dominated under the same relation. If one candidate remains, it is committed, if several remain, the planner selects among them using their verified diagnostics and costs. If none is eligible, the retained plan remains unchanged. Planning terminates upon secure completion or exhaustion of the round budget. Componentwise acceptance enforces monotone progress in the recorded security state, but can exclude sequences that require a temporary increase in one component.

4.3 Consolidating Experience into Reusable Strategies

SCOPE maintains two complementary external memory banks at different levels of abstraction. **(i) Episodic memory** $\mathcal{M}_{\mathrm{epi}}$ stores concrete records of previously completed planning episodes, while **(ii) Procedural memory** $\mathcal{M}_{\mathrm{proc}}$ stores reusable strategies distilled from their planning trajectories. Thus, episodic memory preserves what happened in a particular scenario, whereas procedural memory abstracts how the planner may respond to recurring planning situations.

Episodic memory. For each completed source episode e, episodic memory $\mathcal{M}_{\mathrm{epi}}$ stores its scenario identifier, initial verified state, final plan, and retained-state transition summaries:

$$m_e = \left(s_e, z_{e,0}, d_{e,0}, \pi^{\mathrm{fin}}_e, C(\pi^{\mathrm{fin}}_e), \left[\left(c^{\star}_{e,k}, z_{e,k+1} - z_{e,k}\right)\right]_{k=0}^{T_e-1}\right), \tag{6}$$

where T_e is the number of planning rounds and $c^{\star}_{e,k}$ denotes the best candidate selected in round k under SCOPE's selection rule. The final plan provides a concrete intervention example, while the transition summaries describe how the retained security state progresses across rounds. Candidate labels identify which proposal was selected but do not reconstruct intermediate actions. Rejected candidate records are not included in the episodic entry.

Procedural memory. Procedural memory $\mathcal{M}_{\mathrm{proc}}$ stores reusable strategies distilled from verified planning trajectories. Each strategy has the fixed structure $\sigma = (\mathrm{name}, \mathrm{trigger}, \mathrm{planning\ steps}, \mathrm{limitations}, \mathrm{evidence\ status}, \mathrm{support})$. The *name* is a short label describing the strategy. The *trigger* specifies the verified planning situation in which the strategy is intended to apply; for example, base-case violations may have been resolved while $N-1$ overloads remain. The *planning steps* describe how the planner should respond to that situation, such as focusing subsequent reinforcement on the remaining contingency bottlenecks rather than revisiting already resolved violations. These steps specify a reusable planning procedure rather than fixed asset-level actions; concrete actions are determined from the current diagnostics. The *limitations* record conditions under which the procedure was ineffective, produced adverse effects, or may not apply. The *evidence status* distinguishes guidance supported by observed simulator responses from hypotheses proposed during consolidation. Finally, *support* records the episodes and rounds from which the strategy was derived, together with their round outcomes and Security5 states used later for retrieval.

For each source episode e, SCOPE extracts an episode-level strategy from its verified round records and retained planner reasoning excerpts ρ_e:

$$\sigma_e = \text{Extract}\left(\{\mathcal{R}_{e,k}\}_{k=0}^{T_e-1}, \rho_e\right). \tag{7}$$

The extraction operator is implemented by a consolidation LLM with a fixed, structured prompt. Its input summarizes, for each round, the retained state, the tested candidate updates and their verified outcomes, and the candidate selected at the end of the round, together with the retained reasoning excerpts. The consolidation LLM relates proposed interventions to their verified physical effects and fills the fields of σ_e. Selected transitions provide evidence about interventions that were retained, while informative rejected alternatives can reveal partial improvements, adverse effects, or limitations. Simulator records establish the physical outcomes, while the reasoning excerpts provide context for why the interventions were proposed.

The episode-level strategy is then consolidated with the existing procedural bank:

$$\mathcal{M}_{\text{proc}} \leftarrow \text{Merge}\left(\mathcal{M}_{\text{proc}}, \sigma_e\right). \tag{8}$$

The merge operator compares the trigger and planning steps of σ_e with those of the existing strategies. If an existing strategy describes the same planning situation and recommends the same planning procedure, the two entries are merged while preserving the supporting episode and round references from both. Otherwise, σ_e is added as a new strategy. A merged strategy can therefore retain multiple historical situations that support the same reusable guidance. The complete extraction and merging procedure is given in Algorithm 2 in the appendix.

4.4 Retrieving and Applying Experience

SCOPE retrieves episodic and procedural experience at different stages of planning. Episodic retrieval provides an initial historical precedent, while procedural retrieval supplies guidance conditioned on the evolving verified state.

Episodic retrieval. Before the first planning round, SCOPE compares the current initial Security5 vector $\mathbf{z}_0$ with the initial vectors $\mathbf{z}_{e,0}$ stored in $\mathcal{M}_{\text{epi}}$ for episodes from other scenarios, and retrieves the closest episode $e^\star = \arg\min_{e:\, s_e \neq s} \|\mathbf{z}_{e,0} - \mathbf{z}_0\|_2$. The retrieved episodic entry is provided only to Candidate A as a concrete planning precedent. Its stored intervention is treated as evidence to be adapted using the current diagnostics rather than copied directly, and any resulting proposal is verified on the current scenario in the same way as the other candidates. If no eligible episodic entry is available, Candidate A is generated from the current diagnostics alone.

Procedural retrieval. After a planning round that commits a candidate but leaves violations, finds no eligible candidate, or rejects a candidate with mixed effects, SCOPE retrieves procedural guidance. For each strategy, its *support* contains historical round outcomes and Security5 states. SCOPE first considers supporting rounds with the same outcome as the current round, then scores each strategy by the Euclidean distance between the current Security5 vector $\mathbf{z}_{k+1}$ and its closest matching stored state. This matches strategies to both the type of planning outcome and the remaining violation severity.

The two closest strategies are provided to all candidates in the next round. Their *triggers* indicate whether the guidance applies, and their *planning steps* are instantiated using the current diagnostics. If no strategy has support with the same round outcome, planning continues from the current diagnostics alone. A concrete example of the stored support, merging, and resulting procedural guidance is given in Appendix E.

5 Experiments

We evaluate whether learning from verified planning experience helps SCOPE obtain secure reinforcement plans with lower investment cost and less planning effort. We first compare overall planning performance, then examine the contributions of search and memory. Finally, we analyze what strategies are learned and how they are applied to subsequent scenarios.

5.1 Experimental Setup

Network, scenarios, and model. We use the IEEE 118-bus system augmented with 14 renewable sites (Appendix A.1) and 100 one-day operating scenarios, 20 for each of five conditions: normal, summer, cloudy, storm, and heat wave (Appendix A.2). All configurations use Qwen3.5-9B served with vLLM, the same initial network, actions, and security checks, and at most 30 planning rounds per episode, the remaining settings are listed in Appendix G.

Compared configurations. Each configuration adds one component to the previous one. **(i) Single** proposes one candidate per round, **(ii) Siblings** proposes the three siblings A, B, and C, **(iii) Plan Fusion** adds the fused candidate D, **(iv) Episodic Mem.** adds episodic retrieval for Candidate A. Finally, **(v) SCOPE (full)** additionally uses procedural strategies consolidated from verified planning records.

Five-fold evaluation of experience reuse. The scenarios are split into five groups of 20, each containing four scenarios of every condition. Each scenario is evaluated with memories built only from the other four groups. To build these memories, the 80 scenarios of those groups are first planned with Plan Fusion, which uses no memory, and the resulting records are stored and consolidated. The memories then stay fixed while the group is evaluated. Every scenario is therefore planned once by each configuration, and no scenario ever retrieves its own experience.

Metrics. We report the **(i) Secure completion rate** the percentage of scenarios that end with a secure plan, **(ii) investment cost** of the final plan, **(iii) Planning rounds** the number of rounds needed, **(iv) Completion time** including LLM inference and verification and **(v) Output tokens** generated by the planner. Costs, rounds, time, and tokens are medians over securely completed episodes.

5.2 Overall Planning Performance

Table 1: Overall planning performance on 100 scenarios. Secure completion is over all scenarios; the other metrics are medians over securely completed episodes (99 for Single, 100 otherwise).

Configuration	Secure	Cost ($M)	Rounds	Time (min)	Output tokens
Single	99%	71.60	4	1.51	4,495
Siblings	100%	49.15	4	5.25	29,057
+ Plan Fusion	100%	49.35	4	6.98	35,036
+ Episodic Mem.	100%	51.25	2	3.90	18,940
SCOPE (full)	100%	49.55	2	3.33	14,561

Table 1 compares the five configurations. Compared with the same search without memory (Plan Fusion), SCOPE lowers the mean investment cost by $9.19M per scenario, halves the median number of planning rounds from four to two, and reduces completion time by 52% and output tokens by 58%, while all configurations with sibling search remain secure on every scenario.

5.3 Analysis

Progress over planning rounds. Memory mainly shortens the search. Plan Fusion, which uses no memory, secures 32 of the 100 scenarios after two rounds and 58 after four, SCOPE secures 61 and 90, respectively, and all 100 within eight rounds. Single and Siblings still leave 13 and 8 scenarios unsecured after ten rounds.

What plan fusion contributes. Plan fusion's benefit is concentrated in the more difficult scenarios. Compared with Siblings, adding fusion reduces the median investment cost on summer scenarios from $127.7M to $104.9M. It also reduces the tail of the search distribution: 90 scenarios reach security within eight rounds and 97 within ten, compared with 85 and 92 without fusion, while the longest episode decreases from 19 to 13 rounds. These results suggest that fusion is most useful when no single sibling provides a sufficiently effective intervention, allowing complementary actions proposed by different siblings to be combined and separately verified.

Cost under a matched planning budget. The default planner stops when a secure plan is obtained, although additional rounds can be used to further reduce investment cost. Following the same refinement procedure for all configurations, we measure refinement relative to the point where security is first achieved. Single, Siblings, and Plan Fusion require a median of four planning rounds, while Episodic Mem. and SCOPE require two. Thus, under a common budget of ten total rounds, the

former receive six refinement rounds and the latter receive eight. Each refinement round proposes up to three edits that remove an action, lower a target, or replace an action with a cheaper alternative; every edit is verified, and the cheapest edit that preserves security is retained.

Table 2: Median investment cost ($M) under matched total planning-round budgets. Each row begins at that configuration's median round to secure completion, after which the remaining rounds are used for cost refinement. A dash denotes a round before the aligned post-security trajectory begins.

	Total planning round								
Configuration	2	3	4	5	6	7	8	9	10
Single	–	–	71.60	69.10	68.60	68.10	67.60	67.50	67.00
Siblings	–	–	49.15	48.40	47.90	47.20	46.20	45.45	45.08
+ Plan Fusion	–	–	49.35	49.25	48.05	47.15	46.85	46.60	46.25
+ Episodic Mem.	51.25	50.65	49.90	49.40	47.90	47.53	47.15	46.81	46.06
SCOPE (full)	49.55	49.30	48.25	47.50	46.75	46.20	45.70	45.08	44.33

Table 2 makes the difference in available refinement budget explicit. Episodic Memory and SCOPE reach secure completion at a median of round 2 and can therefore use up to eight additional rounds within a ten-round budget, whereas Single, Siblings, and Plan Fusion begin refinement at round 4 and receive six. At the first round where all configurations are aligned, round 4, SCOPE has a median investment cost of $48.25M, compared with $49.15M for Siblings and $49.35M for Plan Fusion. The costs continue to decrease with additional refinement. At round 10, SCOPE reaches $44.33M, compared with $45.08M for Siblings, $46.25M for Plan Fusion, and $46.06M for Episodic Memory.

5.4 How Memory Helps

Episodic memory. Episodic retrieval provides Candidate A with a concrete precedent before the first planning round. Compared with Plan Fusion, adding episodic memory reduces the median planning rounds from four to two, completion time from 6.98 to 3.90 minutes, and output tokens from 35,036 to 18,940, while maintaining 100% secure completion.

Procedural memory. Procedural memory complements this initial precedent with guidance conditioned on the verified state reached during planning. Across the 100 SCOPE episodes, strategies were retrieved 142 times in 73 episodes. In 138 of these retrievals, at least one candidate in the following round improved on the retained plan, and in 74 cases the committed candidate made the plan secure. Relative to episodic memory alone, SCOPE retains the two-round median while reducing median cost from $51.25M to $49.55M, completion time from 3.90 to 3.33 minutes, and output tokens from 18,940 to 14,561. These results suggest complementary roles for the two memories: episodic retrieval provides an initial planning precedent, while procedural retrieval supplies reusable guidance as the remaining violations evolve. Representative successful and unsuccessful reuse cases are given in Appendix F.

6 Conclusion

We introduced SCOPE, a self-evolving LLM agent for secure power grid reinforcement planning that learns from verified planning experience without updating its parameters. Within each episode, SCOPE generates complementary candidate plans from a common retained state, fuses their actions into an additional candidate, and commits only improvements verified by AC power flow and $N-1$ contingency analysis. Across episodes, it stores concrete precedents in episodic memory and consolidates verified planning records into reusable, state-conditioned strategies in procedural memory. On 100 operating scenarios, SCOPE secured every scenario and, compared with the same search without memory, halved the median number of planning rounds and reduced completion time and output tokens by 52% and 58%, respectively. At the same total number of rounds, SCOPE also reached the lowest median investment cost after cost refinement. We chose the IEEE 118-bus system because it is large enough to test SCOPE, its network of 186 lines and transformers, combined with five operating conditions, produces base-case and $N-1$ violations that require coordinated reinforcement, while every candidate can still be verified with full AC power flow. In future work, we will extend SCOPE to various power grid systems.

AI USE STATEMENT

In this work, we used generative AI tools to assist with identifying potentially relevant prior work during the early stages of the research process. We independently read, reviewed, and evaluated the identified literature. We also used generative AI tools during the writing process to improve grammar and sentence clarity. We carefully reviewed and verified all AI-assisted changes and take full responsibility for the final text and all other content of this work.

ETHICS STATEMENT

This work does not raise any ethical concerns that require further clarification.

REPRODUCIBILITY STATEMENT

The complete source code is available at `https://anonymous.4open.science/r/SCOPE-5BAD/`

REFERENCES

S Babaeinejadsarookolaee, A Birchfield, RD Christie, C Coffrin, C DeMarco, R Diao, M Ferris, S Fliscounakis, S Greene, R Huang, et al. The power grid library for benchmarking ac optimal power flow algorithms. arxiv. *arXiv preprint arXiv:1908.02788*, 2019.

Xin Chen et al. X-gridagent: An llm-powered agentic ai system for assisting power grid analysis. *arXiv preprint arXiv:2512.20789*, 2025.

Runnan Fang, Yuan Liang, Xiaobin Wang, Jialong Wu, Shuofei Qiao, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. Memp: Exploring agent procedural memory. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 17490–17502, 2026.

Hongwei Jin, Kibaek Kim, and Jonghwan Kwon. Gridmind: Llms-powered agents for power system analysis and operations. In *Proceedings of the SC'25 Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 560–568, 2025.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T Le, Samira Daruki, Xiangru Tang, et al. Reasoningbank: Scaling agent self-evolving with reasoning memory, 2025. *URL https://arxiv. org/abs/2509.25140*.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. *URL https://arxiv. org/abs/2303.11366*, 1:5, 2023.

L. Thurner, A. Scheidler, F. Schäfer, J. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun. pandapower — an open-source python tool for convenient modeling, analysis, and optimization of electric power systems. *IEEE Transactions on Power Systems*, 33(6):6510–6521, Nov 2018. ISSN 0885-8950. doi: 10.1109/TPWRS.2018.2829021.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models, 2023. *URL https://arxiv. org/abs/2305.16291*, 2(11), 2023.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. *arXiv preprint arXiv:2409.07429*, 2024.

Rong Wu, Xiaoman Wang, Jianbiao Mei, Pinlong Cai, Daocheng Fu, Cheng Yang, Licheng Wen, Xuemeng Yang, Yufan Shen, Yuxin Wang, et al. Evolver: Self-evolving llm agents through an experience-driven lifecycle, 2025. *URL https://arxiv. org/abs/2510.16079*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, et al. Aflow: Automating agentic workflow generation. In *International Conference on Learning Representations*, volume 2025, pp. 34040–34077, 2025a.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, et al. Agentic context engineering: Evolving contexts for self-improving language models. doi: 10.48550. *arXiv preprint arXiv.2510.04618*, 2025b.

Shengtao Zhang, Jiaqian Wang, Ruiwen Zhou, Junwei Liao, Yuchen Feng, Zhuo Li, Yujie Zheng, Weinan Zhang, Ying Wen, Zhiyu Li, et al. Memrl: Self-evolving agents via runtime reinforcement learning on episodic memory. *arXiv preprint arXiv:2601.03192*, 2026.

Yan Zhang, Ahmad Mohammad Saber, Amr Youssef, and Deepa Kundur. Grid-agent: An llm-powered multi-agent system for power grid control. *arXiv preprint arXiv:2508.05702*, 2025c.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models, 2024. *URL https://arxiv.org/abs/2310.04406*, 4.

A Network, Scenarios, and Physical Model

A.1 Test Network

We use the IEEE 118-bus system in its PGLib-OPF version (Babaeinejadsarookolaee et al., 2019) (v23.07, base 100 MVA), augmented with renewable generation. It has $|\mathcal{B}| = 118$ buses, 175 transmission lines, and 11 transformers, so that $|\mathcal{E}^0| = 186$; branch ratings are the PGLib long-term ratings. The network contains 99 loads, 54 conventional generators, 14 existing bus shunts, and 14 added renewable sites, namely 6 wind and 8 solar sites, located at the buses listed in Table 3. Of the 175 lines, 165 operate at 138 kV and 10 at 345 kV.

Table 3: Renewable sites added to the IEEE 118-bus system.

Type	Buses
Wind	10, 26, 59, 65, 90, 100
Solar	16, 34, 49, 70, 80, 92, 105, 116

A.2 Operating Scenarios

Generation. We generate a synthetic dataset of one-day operating scenarios for five conditions. Each scenario specifies, for every hour $h \in \mathcal{H} = \{1, \ldots, 24\}$, the active demand of each load, the available active power of each wind and solar site, and the availability of each conventional generator, and is generated with its own random seed. The conditions differ as follows (Table 4):

- **Normal:** reference demand and renewable output.
- **Summer:** the highest demand, with a median peak about 28% above normal, combined with low wind output.
- **Cloudy:** demand close to normal, with slightly reduced solar and wind output.
- **Storm:** demand close to normal, less than half of the normal solar output, and strongly fluctuating wind output.
- **Heat wave:** high demand, the largest hour-to-hour demand changes, and reduced wind output.

Scenarios contain no generator or line outages, so outages enter only through the $N-1$ analysis, and every conventional unit is available in every hour. Reactive demand follows each load's baseline ratio, renewable sites inject no reactive power, and dispatch follows the rule in Appendix A.4. We use 100 scenarios, 20 per condition, which were fixed before any experiment reported here and are not reselected based on outcomes.

Table 4: Scenario conditions (20 scenarios each). Values are medians over the scenarios of a condition; wind and solar are daily energies relative to the normal condition.

	Demand (MW)		Renewables	
Condition	Peak	Min.	Wind	Solar
Normal	4032	2213	100%	100%
Summer	5159	2832	53%	107%
Cloudy	4040	2224	95%	96%
Storm	4067	2228	88%	44%
Heat wave	4731	2454	62%	103%

No scenario is secure without reinforcement. All 100 scenarios have $N-1$ overloads and unacceptable contingencies; 80 have $N-1$ voltage violations, 34 base-case overloads, and 18 base-case voltage violations.

Evaluation groups. For the five-fold evaluation of experience reuse, the scenarios are divided into five groups of 20, each containing four scenarios of every condition (Table 5). Within each condition, scenarios are numbered 01–20 in the fixed evaluation ordering, and consecutive blocks of four form groups 1 to 5. Memories for a group are built only from the other four groups, and runtime retrieval additionally excludes the current scenario.

Table 5: Evaluation groups. Scenarios are numbered 01–20 within each condition.

Group	Normal	Summer	Cloudy	Storm	Heat wave
1	01, 02, 03, 04	01, 02, 03, 04	01, 02, 03, 04	01, 02, 03, 04	01, 02, 03, 04
2	05, 06, 07, 08	05, 06, 07, 08	05, 06, 07, 08	05, 06, 07, 08	05, 06, 07, 08
3	09, 10, 11, 12	09, 10, 11, 12	09, 10, 11, 12	09, 10, 11, 12	09, 10, 11, 12
4	13, 14, 15, 16	13, 14, 15, 16	13, 14, 15, 16	13, 14, 15, 16	13, 14, 15, 16
5	17, 18, 19, 20	17, 18, 19, 20	17, 18, 19, 20	17, 18, 19, 20	17, 18, 19, 20

A.3 Reinforcement Actions and Costs

An action $u = (a, \xi, v) \in \mathcal{U}$ applies operation $a \in \mathcal{A}$ to the asset or location ξ with the *final* target value v, not an increment. For an existing line or transformer ξ, let S^0_ξ denote its original rating in MVA. The four operations and the way each changes the simulator model are as follows.

- **Line upgrade.** ξ is an existing line and $v \geq S^0_\xi$ is its final rating in MVA. The upgrade models reconductoring: with $\kappa = v/S^0_\xi$, the thermal current limit is multiplied by κ, the resistance is divided by κ, and the reactance is divided by $\kappa^{0.3}$.
- **Transformer upgrade.** ξ is an existing transformer and $v \geq S^0_\xi$ is its final rating in MVA. The short-circuit voltages, which are given relative to the rating, are rescaled with it, so that the transformer's impedance in ohms is unchanged.
- **New-line construction.** $\xi = (i, j)$ is a pair of existing buses and $v \geq 0$ is the rating of the new circuit in MVA. The new circuit is a complete AC branch between buses i and j. If a line of the PGLib network already connects these buses, the new circuit is a parallel circuit and copies that line's resistance and reactance. Otherwise, it receives $r = 3\,(V_i/138)^2\,\Omega$ and $x = 12\,(V_i/138)^2\,\Omega$, where V_i is the nominal voltage of bus i in kV, which corresponds to $r = 0.016$ and $x = 0.063$ p.u. on the 100 MVA base. Like all lines in the model, the new circuit has no shunt charging, and its thermal current limit is $v/(\sqrt{3}\,V_i)$. A plan contains

at most one new circuit per bus pair; a later action on the same pair revises its rating, and $v = 0$ removes it.

- **Reactive support.** ξ is a bus and $v \in \mathbb{R}$ is the final setpoint in MVAr, modeled as a constant-susceptance shunt that injects v at 1 p.u. voltage (capacitive if $v > 0$, inductive if $v < 0$). It is added to any existing shunt at that bus, and $v = 0$ removes previously planned support.

The admissible action set $\mathcal{U}$ also incorporates topology-derived bounds on reinforcement targets. Let $\mathcal{L}^{\mathrm{ex}}$ denote the existing transmission lines and $\mathcal{T}$ the existing transformers.

For an existing line $\ell \in \mathcal{L}^{\mathrm{ex}}$, let ν_ℓ denote its nominal voltage class and define

$$\overline{S}^{L}(\nu) = \max_{\substack{e \in \mathcal{L}^{\mathrm{ex}} \\ \nu_e = \nu}} S_e^0. \tag{9}$$

The final target of a line upgrade must therefore satisfy

$$S_\ell^0 \le S_\ell^{\mathrm{new}} \le \overline{S}^{L}(\nu_\ell). \tag{10}$$

Thus, an existing line can be reinforced only up to the largest original rating represented by a line of the same voltage class.

For a transformer $\tau \in \mathcal{T}$, let c_τ denote its transformer class, defined by its nominal high and low voltage levels, and define

$$\overline{S}^{T}(c) = \max_{\substack{r \in \mathcal{T} \\ c_r = c}} S_r^0. \tag{11}$$

A transformer target must satisfy

$$S_\tau^0 \le S_\tau^{\mathrm{new}} \le \overline{S}^{T}(c_\tau). \tag{12}$$

New-line construction is restricted to a predefined candidate-corridor set $\mathcal{L}^{\mathrm{cand}}$. Hence a new-line action may be instantiated only when

$$(i, j) \in \mathcal{L}^{\mathrm{cand}}. \tag{13}$$

For each candidate corridor, the new circuit uses the rating of its reference circuit,

$$S_{ij}^{\mathrm{new}} = S_{ij}^{\mathrm{ref}}, \tag{14}$$

and at most one additional circuit may be constructed:

$$n_{ij}^{N} \in \{0, 1\}. \tag{15}$$

Reactive-support targets are bounded using the scale of shunt equipment already represented in the network. Let $\mathcal{H}(\nu)$ denote the set of original shunt devices connected at voltage class ν, and define

$$\overline{Q}^{\mathrm{sh}}(\nu) = \max_{h \in \mathcal{H}(\nu)} \left| Q_h^0 \right|. \tag{16}$$

For a candidate bus i with nominal voltage class ν_i, the final reactive-support target satisfies

$$-\overline{Q}^{\mathrm{sh}}(\nu_i) \le Q_i^{\mathrm{sh}} \le \overline{Q}^{\mathrm{sh}}(\nu_i). \tag{17}$$

If no reference shunt exists at a voltage class, its admissible bound is specified as a benchmark parameter.

The set $\mathcal{U}$ contains the instantiated actions satisfying these operation-specific domains. A reinforcement plan $\pi \in \Pi$ is a finite collection of compatible actions from $\mathcal{U}$, with no conflicting final targets assigned to the same asset or installation location. The branch set after applying π is

$$\mathcal{E}(\pi) = \mathcal{E}^0 \cup \left\{ (i, j) : u \in \pi,\ a_u = \text{new-line construction},\ n_{ij}^{N} = 1 \right\}. \tag{18}$$

The cost of an instantiated action $u = (a, \xi, v)$, in millions of US dollars, is

$$\gamma(u) = \begin{cases} 2.0 + 0.05\,(v - S_\xi^0), & a = \text{line upgrade},\ v > S_\xi^0, \\ 2.5 + 0.035\,(v - S_\xi^0), & a = \text{transformer upgrade},\ v > S_\xi^0, \\ 20.0 + 0.10\,v, & a = \text{new-line construction},\ v > 0, \\ 0.5 + 0.02\,|v|, & a = \text{reactive support},\ v \neq 0, \\ 0, & \text{otherwise}. \end{cases} \tag{19}$$

so an action that restores the original state costs nothing, and $C(\pi) = \sum_{u \in \pi} \gamma(u)$. Because each target is a final value, the cost of an upgrade depends only on the difference between the final and the original rating, regardless of how many times the target was revised during planning. For example, upgrading a line from 100 to 150 MVA costs $2.0 + 0.05 \cdot 50 = 4.5$ million dollars, and a new 200 MVA line costs $20.0 + 0.10 \cdot 200 = 40.0$ million dollars. These are stylized planning costs chosen for this benchmark rather than market estimates. Table 6 summarizes the operations.

Table 6: Reinforcement operations. Costs are in millions of US dollars and are zero outside the stated condition; ratings are in MVA and reactive support in MVAr.

Operation a	Asset ξ	Target v	Also changes	Cost $\gamma(u)$
Line upgrade	existing line	rating $\geq S^0_\xi$	resistance, reactance	$2.0 + 0.05\,(v - S^0_\xi)$
Transformer upgrade	existing transformer	rating $\geq S^0_\xi$	short-circuit voltages	$2.5 + 0.035\,(v - S^0_\xi)$
New-line construction	bus pair (i, j)	rating ≥ 0	adds a branch	$20.0 + 0.10\,v$
Reactive support	bus	setpoint $\in \mathbb{R}$	bus shunt susceptance	$0.5 + 0.02\,\lvert v \rvert$

A.4 Bus Injections and AC Power-Flow Equations

Let $\mathcal{D}$ be the set of loads and $\mathcal{R}$ the set of renewable sites, and let $\mu^L : \mathcal{D} \to \mathcal{B}$ and $\mu^R : \mathcal{R} \to \mathcal{B}$ map each load and each site to its bus. A scenario specifies the active demand $P^L_{d,h}$ of every load and the available active power $P^{R,\mathrm{av}}_{r,h}$ of every site. The reactive demand of load d preserves its baseline ratio, $Q^L_{d,h} = P^L_{d,h}\, Q^{L,0}_d / P^{L,0}_d$, where $P^{L,0}_d$ and $Q^{L,0}_d$ are its baseline demands, and each renewable site injects its available active power and no reactive power. The bus quantities are

$$P^L_{i,h} = \sum_{d:\, \mu^L(d)=i} P^L_{d,h}, \qquad Q^L_{i,h} = \sum_{d:\, \mu^L(d)=i} Q^L_{d,h}, \qquad P^R_{i,h} = \sum_{r:\, \mu^R(r)=i} P^{R,\mathrm{av}}_{r,h}. \tag{20}$$

For the network reinforced by π, with branch $o \in \mathcal{E}(\pi)$ removed in a contingency (or no branch removed in the base case), a converged operating point at hour h satisfies, for every bus $i \in \mathcal{B}$,

$$\begin{aligned} P^G_{i,h} + P^R_{i,h} - P^L_{i,h} &= V_{i,h} \sum_{j \in \mathcal{B}} V_{j,h} \big[G_{ij} \cos(\theta_{i,h} - \theta_{j,h}) + B_{ij} \sin(\theta_{i,h} - \theta_{j,h}) \big], \\ Q^G_{i,h} - Q^L_{i,h} &= V_{i,h} \sum_{j \in \mathcal{B}} V_{j,h} \big[G_{ij} \sin(\theta_{i,h} - \theta_{j,h}) - B_{ij} \cos(\theta_{i,h} - \theta_{j,h}) \big], \end{aligned} \tag{21}$$

where $P^G_{i,h}$ and $Q^G_{i,h}$ are the active and reactive output of the conventional generators at bus i, $V_{i,h}$ and $\theta_{i,h}$ are the voltage magnitude and angle of bus i, and G_{ij} and B_{ij} are the entries of the bus admittance matrix of the reinforced network. This matrix is built from the branch set $\mathcal{E}(\pi)$ and the electrical parameters set by π (Appendix A.3); it includes the existing shunts and the planned reactive support as constant susceptances, and lines are modeled by their series impedance without shunt charging. The unit at bus 69 acts as the slack bus with a voltage setpoint of 1.0 p.u., and the other units are voltage-controlled at their setpoints. In each hour, the units share the net load, i.e., total demand minus renewable output, in proportion to their base-case output and within their limits, and the slack unit supplies the remaining imbalance and losses. Each solve uses the Newton–Raphson method of `pandapower` with a mismatch tolerance of 10^{-8} MVA and at most 50 iterations, without enforcing generator reactive-power limits.

A.5 Security5 Computation

Notation. Let $\mathcal{H}^{\mathrm{conv}}(\pi, s) \subseteq \mathcal{H}$ denote the base-case hours whose AC power flow converges, and let $\mathcal{C}^{\mathrm{conv}}(\pi, s)$ denote the peak-hour contingencies whose AC power flow converges. Let $\rho_{\ell,h}$ and $V_{i,h}$ be the loading ratio of branch ℓ and voltage magnitude of bus i in base-case hour h, and let $\rho_{\ell,c}$ and $V_{i,c}$ denote the corresponding quantities for a converged contingency c. We write $[x]_+ = \max(x, 0)$. The peak hour is the hour of maximum total active demand.

The RMS normalizers count the valid observations that actually enter each component: N_{BT} is the number of valid base-case branch–hour observations, N_{BV} the number of valid base-case bus–hour voltage observations, N_{NT} the number of valid surviving-branch observations over converged contingencies, and N_{NV} the number of valid bus-voltage observations over converged contingencies.

Base-case thermal severity. For a branch loading ratio $\rho_{\ell,h}$,

$$\phi_{T,B}(\rho_{\ell,h}) = [\rho_{\ell,h} - 1]_+, \tag{22}$$

and

$$T_B = \left(\frac{1}{N_{BT}} \sum_{h \in \mathcal{H}^{\text{conv}}} \sum_{\ell \in \mathcal{E}(\pi)} \phi_{T,B}(\rho_{\ell,h})^2 \right)^{1/2}, \tag{23}$$

where only valid finite observations are included in the sum.

Base-case voltage severity. For bus voltage $V_{i,h}$,

$$\phi_{V,B}(V_{i,h}) = \max \left\{ \frac{[0.95 - V_{i,h}]_+}{0.95}, \frac{[V_{i,h} - 1.05]_+}{1.05} \right\}, \tag{24}$$

and

$$V_B = \left(\frac{1}{N_{BV}} \sum_{h \in \mathcal{H}^{\text{conv}}} \sum_{i \in \mathcal{B}} \phi_{V,B}(V_{i,h})^2 \right)^{1/2}. \tag{25}$$

$N-1$ **thermal severity.** For a surviving branch ℓ under a converged contingency c,

$$\phi_{T,N1}(\rho_{\ell,c}) = \frac{[\rho_{\ell,c} - 1.10]_+}{1.10}, \tag{26}$$

and

$$T_{N1} = \left(\frac{1}{N_{NT}} \sum_{c \in \mathcal{C}^{\text{conv}}} \sum_{\ell \in \mathcal{E}(\pi) \setminus \{c\}} \phi_{T,N1}(\rho_{\ell,c})^2 \right)^{1/2}, \tag{27}$$

where the outaged branch is not included in its own contingency observations.

$N-1$ **voltage severity.** The current contingency screen uses only the emergency lower-voltage limit. Thus,

$$\phi_{V,N1}(V_{i,c}) = \frac{[0.92 - V_{i,c}]_+}{0.92}, \tag{28}$$

and

$$V_{N1} = \left(\frac{1}{N_{NV}} \sum_{c \in \mathcal{C}^{\text{conv}}} \sum_{i \in \mathcal{B}} \phi_{V,N1}(V_{i,c})^2 \right)^{1/2}. \tag{29}$$

$N-1$ **failure fraction.** Let N_{tested} be the number of peak-hour contingencies tested and N_{bad} the number classified as unacceptable. A contingency is unacceptable if its AC power flow does not converge, a surviving branch exceeds 110% loading, or a bus voltage falls below 0.92 p.u. Then

$$F_{N1} = \frac{N_{\text{bad}}}{N_{\text{tested}}}. \tag{30}$$

Every observation within its corresponding limit contributes zero to the RMS. For example, base-case branch loadings of 90%, 105%, and 150% have element-level normalized thermal violations 0, 0.05, and 0.50, respectively, before RMS aggregation. Base-case voltages of 0.94 and 0.80 p.u. have normalized low-voltage violations $0.01/0.95 \approx 0.0105$ and $0.15/0.95 \approx 0.158$. A branch loaded at 121% under a contingency has normalized thermal violation $0.11/1.10 = 0.10$. If 20 of 100 tested contingencies are unacceptable, then $F_{N1} = 0.20$.

New lines. A new line enlarges $\mathcal{E}(\pi)$: it is monitored in each base-case hour and in the other contingency cases, and its own outage is also tested. Hence the observation counts N_{BT}, N_{NT} and the number of tested contingencies can change with the plan, and all five Security5 components are recomputed for every evaluated candidate.

Non-converged solves. A non-converged solve provides no valid voltages or loadings, so it contributes no observation to the corresponding RMS sums. A non-converged contingency is nevertheless classified as unacceptable and therefore contributes to F_{N1}. Non-converged base-case hours are returned separately in the verifier diagnostics.

Diagnostics. Together with Security5, the verifier returns the non-converged base-case hours, the numbers of base-case problem hours and unacceptable contingencies, and the overloaded branches and out-of-limit buses observed in the base case and under contingencies, together with their worst loadings and voltages. These diagnostics are supplied to the planner.

Example of the preference rule. Under Eq. (5), a candidate that changes the security state from $[0.20, 0.05, 0.30, 0.08, 0.25]$ to $[0.08, 0.03, 0.12, 0.04, 0.10]$ improves every component and is preferred to its parent. A candidate that reaches $[0.05, 0.07, 0.10, 0.02, 0.05]$ from the same parent reduces four components but increases V_B, so it is not preferred to the parent and is not eligible for commitment.

B Planning and Refinement Details

Candidate validation and repair. The three sibling updates are submitted in one tool call and verified one at a time from the same retained plan. Before verification, each update is checked: every action must refer to an existing line, transformer, bus, or pair of buses and satisfy the target domain of its operation (Appendix A.3); each update must contain at least one action that changes the retained plan; and the three resulting plans must differ pairwise in at least one asset, operation, or effective target value. Actions that change nothing, such as a target equal to the current value, are reported and do not count toward this difference. The restriction of new-line construction to Candidate C is given by the role prompts (Appendix D). If a proposal violates these rules, the tool returns the specific error and the planner repairs the proposal in a separate call (Table 7). The repaired siblings are then verified.

Plan Fusion validation. After the siblings are verified, their effective actions are listed in an action catalog in which each action has a source reference such as `A:1` or `C:3`. The planner builds D by selecting references from this catalog. The tool accepts D only if every reference exists, at most one target is selected per asset, and the references come from at least two siblings; the selected targets are kept unchanged. A rejected fusion is returned to the planner with the specific error, and the planner corrects it.

Selection among several eligible candidates. Eligible candidates that are dominated under Eq. (5) by another eligible candidate are removed. If exactly one candidate remains, it is committed. If several remain, each better in a different Security5 component, the planner receives their verified Security5 vectors, diagnostics, and costs and commits one of them.

Cost refinement. The refinement stage of Section 5.3 starts from the final secure plan of each scenario and configuration, which is first verified again. In each round, the planner receives the current plan, the cost and original rating of every action, and all edits tested so far with their outcomes, and proposes up to three edits. Each edit changes exactly one action: it removes the action, lowers its target on the same asset, or replaces it with one cheaper action on another asset. Edits that are invalid, that do not lower the plan cost, or that repeat an earlier edit are discarded. Every remaining edit is verified with the same AC power-flow and $N-1$ checks as during planning, and the cheapest edit whose plan is secure is adopted; otherwise the plan is kept. All configurations use the same number of rounds and edits per round, and the final plan is verified once more.

C Memory Construction and Retrieval

C.1 Memory Construction

Algorithm 2 constructs the two memories for one evaluation group from the completed source episodes in the other four groups. These source episodes are generated by Plan Fusion without memory retrieval.

Algorithm 2 Building episodic and procedural memories from source episodes

Require: verified records $\{\mathcal{R}_{e,k}\}_{k=0}^{T_e-1}$ and retained reasoning excerpts ρ_e for source episodes $e \in \mathcal{E}_{\text{src}}$

$\mathcal{M}_{\text{epi}} \leftarrow \emptyset$, $\mathcal{M}_{\text{proc}} \leftarrow \emptyset$
for each source episode $e \in \mathcal{E}_{\text{src}}$ **do**
  Construct m_e by Eq. (6) and add it to $\mathcal{M}_{\text{epi}}$
  Build a compact state ledger from $\{\mathcal{R}_{e,k}\}_{k=0}^{T_e-1}$
  $\sigma_e \leftarrow \text{Extract}(\{\mathcal{R}_{e,k}\}_{k=0}^{T_e-1}, \rho_e)$ by Eq. (7)
  $\mathcal{M}_{\text{proc}} \leftarrow \text{Merge}(\mathcal{M}_{\text{proc}}, \sigma_e)$ by Eq. (8)
end for
return $\mathcal{M}_{\text{epi}}$ and $\mathcal{M}_{\text{proc}}$

Episodic entries. For each completed source episode e, the episodic entry follows Eq. (6). It stores the scenario identifier s_e, the initial verified Security5 state and diagnostics $(z_{e,0}, d_{e,0})$, the final retained plan π_e^{fin} and its cost, and the retained-state transition summaries $\{(c^\star_{e,k}, z_{e,k+1} - z_{e,k})\}_{k=0}^{T_e-1}$. The candidate label identifies the proposal selected under SCOPE's commitment rule in that round, while the Security5 difference records the resulting change in the retained security state. Intermediate candidate plans and rejected alternatives are not reproduced in the episodic entry; their verified outcomes remain available through the full round records used for procedural consolidation.

State ledger for Procedural strategy extraction. The consolidation input is built from the round records $\{\mathcal{R}_{e,k}\}_k$. For each round it summarizes the retained state, every tested candidate update and verified outcome, candidate cost, whether the candidate was eligible and committed, the resulting retained transition and the planner reasoning logs. It also records which Security5 components were cleared, reduced but remained nonzero, or worsened. Section E gives a concrete example.

Extraction. The consolidation LLM receives the state ledger, the retained planner reasoning excerpts in chronological order, and the episode outcome, and returns exactly one episode-level strategy with the fixed fields defined in Section 4.3. The prompt asks for a trigger stated as a planning state rather than as source-specific equipment, planning steps that use current diagnostics, limitations, evidence status, and references to the supporting rounds. Simulator records establish the physical outcomes; retained reasoning supplies context for why the interventions were proposed. Outputs that do not satisfy the structured schema are repaired before they are admitted to the bank.

Merging. The consolidation LLM compares the trigger and planning steps of the new episode-level strategy with the strategies already in $\mathcal{M}_{\text{proc}}$. Strategies that describe the same planning situation and recommend the same planning procedure are merged while preserving the supporting episode and round references from both; otherwise the new strategy is retained as a separate entry. Supporting rounds preserve the round outcome and retained Security5 state later used for retrieval.

C.2 Retrieval Details

Episodic retrieval. Before the first planning round, SCOPE compares the current initial Security5 vector with the initial vectors stored in $\mathcal{M}_{\text{epi}}$, excludes entries from the current scenario, and retrieves the episode with the smallest Euclidean distance. The retrieved entry is provided only to Candidate A. If no eligible entry is available, Candidate A uses the current diagnostics alone.

Procedural retrieval. Procedural guidance is requested after a round that commits a candidate but leaves violations, finds no eligible candidate, or rejects a candidate with mixed effects. Each strategy's support field retains the historical round outcome and Security5 state from which it was learned. SCOPE first considers only supporting rounds with the same outcome as the current round. For each strategy, it then computes the smallest Euclidean distance between the current Security5 vector and the Security5 vectors of these matching supporting rounds. The two strategies with the smallest distances are returned to all candidates in the next round. Their triggers indicate when the guidance applies, while their planning steps are interpreted using the current diagnostics to construct concrete actions. If no strategy has supporting evidence with the same outcome, planning continues from the current diagnostics alone.

D PROMPTS

The planner and the consolidation LLM use fixed, structured prompts.

Candidate roles (candidate generation, one call for A, B, and C):

> **A:** Candidate A is the dedicated episodic-memory candidate. When recall_experience found a historical episode, use it as historical evidence and inspiration rather than as a template to copy. Compare the recalled experience with the CURRENT analyse result. Notice useful equipment choices, intervention patterns, and plausible target ranges, then construct A from the current diagnostics. Historical actions may be reused, retargeted, ignored, or supplemented when current physical evidence supports doing so. If recall_experience found no prior episode, fall back to a coherent current-evidence plan. Do not propose new_line actions.
>
> **B:** Cost-efficient plan. Seek meaningful security improvement with economical investment. Be strategic about which problems to address and which interventions can address several problems together. Prefer a focused action set with justified equipment choices and target sizes. Do not sacrifice security eligibility to obtain a lower cost. Do not propose new_line actions.
>
> **C:** Security-focused plan. Prioritize resolving the current security problems, with the greatest willingness to invest in additional or coordinated reinforcement. C may propose a new_line action only when current physical evidence specifically supports a topology or alternative-path intervention. Do not assume the most expensive plan is the most secure. Let the simulator establish the actual Security5 and total plan cost.
>
> **All three candidates:** start from the SAME retained parent; follow the same Security5 eligibility rules; must satisfy the structural-diversity requirements Every pair must differ in at least one effective equipment selection, action type, or effective target value.

Strategy extraction (one call per source episode):

> Choose exactly ONE strongest reusable PROCEDURAL strategy from this episode. [...] The authoritative episode state ledger determines what actually happened. Planner reasoning explains intent only. [...] The preferred trigger is normally a planning STATE, not a hard-coded intervention. [...] Before the tool call, perform two silent checks: 1. Causal abstraction: Would the strategy still be useful if different equipment achieved the same Security5 state transition? 2. Trajectory faithfulness: Can every major step of the proposed procedure be tied to an observed retained/reverted transition or planner correction in this episode?

Strategy merging (one call per batch of episode strategies):

> Compare strategies by: situation trigger; ordered decision logic; intended planning role; important limitations. Merge strategies ONLY when those semantics are substantially the same. [...] Do not merge merely because both mention: thermal problems; voltage problems; cost; crossover; candidate generation; the same action type. If equivalent strategies are supported by different episodes, merge them and cite all supporting source_keys. [...] Existing bank entries may be retained, refined, merged, or retired when superseded.

Cost refinement (one call per refinement round): the current secure plan is listed with the cost and original rating of each action, together with all edits already tested and their outcomes. The LLM proposes up to three edits, each of which removes one action, lowers the target of one action, or replaces one action with a cheaper action on another asset.

E EXAMPLE OF PROCEDURAL STRATEGY CONSOLIDATION

We give one representative example to illustrate how verified planning records are converted into procedural guidance. The example comes from the procedural memory constructed for evaluation group 1. Its source episode is scenario `storm_15`, which belongs to the source set and is planned by Plan Fusion without memory.

The consolidation procedure receives the verified outcomes of all tested candidates in the episode, including both candidates that were committed and alternatives that were not retained, together

with the retained planner reasoning. This contrast provides evidence about both effective and ineffective interventions. In rounds 3-5, for example, a committed candidate in round 3 reduces the retained Security5 state to $[0, 0, 0.0002, 0, 0.0054]$, clearing the base-case components and V_{N1} while leaving small residual T_{N1} and F_{N1}. In round 4, the tested alternatives do not produce an eligible improvement, so none is committed and the retained state remains unchanged. In round 5, another committed candidate clears the remaining violations, producing $[0, 0, 0, 0, 0]$. The extraction step therefore uses both successful retained transitions and unsuccessful alternatives to infer when a planning procedure is useful and where its limitations arise.

Episode-level strategy.

> **Name:** Target residual thermal and failure bottlenecks with line upgrades.
>
> **Trigger:** the retained plan has cleared voltage severity ($V_{N1} = 0$), but thermal severity (T_{N1}) and/or failure fraction (F_{N1}) remain nonzero.
>
> **Planning steps:** (1) identify the current nonzero Security5 components using diagnostics; (2) when V_{N1} is already zero but T_{N1} or F_{N1} remains nonzero, prioritize line capacity upgrades over reactive support; (3) select the most critical overloaded line(s) based on current thermal loading and contingency analysis; (4) apply targeted line upgrades to clear the remaining thermal or failure bottleneck; and (5) verify that previously cleared components (V_{N1}) remain zero after the upgrade.
>
> **Limitations:** reactive support alone cannot clear thermal severity or failure fraction when they remain nonzero; line upgrades may be required even when voltage is already satisfied; the strategy assumes thermal and failure bottlenecks are the remaining issues, not voltage.
>
> **Evidence status:** observed response.
>
> **Support:** rounds 3–5 of `storm_15`. Their stored round outcomes and retained Security5 states are `after_commit` with $[0, 0, 0.0002, 0, 0.0054]$, `nonimproving_round` with $[0, 0, 0.0002, 0, 0.0054]$, and `after_commit` with $[0, 0, 0, 0, 0]$, respectively.

Strategy after merging. The episode-level strategy is then compared with strategies extracted from the other source episodes. Strategies describing compatible planning situations and planning procedures are merged while their supporting historical evidence is preserved. In the resulting group-1 procedural bank, the strategy above contributes to the following merged entry, supported by 28 source episodes and 61 round-level support records.

> **Name:** Target residual $N-1$ thermal and failure bottlenecks with line upgrades and reactive support.
>
> **Trigger:** the retained plan has cleared T_B and V_B but T_{N1} and/or F_{N1} remain nonzero; current diagnostics show thermal overloads or $N-1$ failure fraction issues.
>
> **Planning steps:** (1) use CURRENT diagnostics to identify overloaded lines in $N-1$ outage tests and voltage problem buses; (2) prioritize line upgrades for lines with highest loading percentages to address thermal severity; (3) add reactive support at critical buses if thermal upgrades alone cannot clear F_{N1}; (4) protect previously cleared components (T_B, V_B, V_{N1}) from regression; and (5) test the combined intervention and recompute retained Security5 to determine if T_{N1} and F_{N1} are cleared or if further refinement is needed.
>
> **Limitations:** reactive support alone cannot clear thermal severity or failure fraction when they remain nonzero; line upgrades may be required even when voltage is already satisfied; new line additions may be required if existing infrastructure cannot resolve F_{N1}; cost efficiency may require balancing multiple intervention types.
>
> **Evidence status:** observed response.
>
> **Support:** 28 source episodes and 61 supporting rounds. Each supporting round retains its round outcome and verified Security5 state rather than being collapsed into a single representative state.

The merged support therefore contains multiple historical Security5 states associated with the same reusable planning procedure. For example, stored supporting states include $[0, 0, 0.0002, 0, 0.0054]$, $[0, 0, 0.0013, 0, 0.0376]$, and $[0, 0, 0.0030, 0, 0.0753]$ from different source episodes. These states are preserved individually during merging rather than averaged. Consequently, the merged strategy can represent the same planning guidance across historical situations with different residual violation magnitudes, while the current diagnostics still determine the concrete reinforcement actions when the strategy is reused.

This example illustrates the two stages of procedural memory construction:

verified round records $\longrightarrow$ episode-level strategy $\longrightarrow$ merged strategy with multi-episode support.

F Examples of Procedural Strategy Reuse

Successful reuse. In `summer_1`, after the base-case violations and V_{N1} were cleared, only T_{N1} and F_{N1} remained. Over the next four rounds, SCOPE repeatedly retrieved guidance for addressing residual $N-1$ thermal and failure bottlenecks using the current diagnostics. The retained plans progressively reduced the remaining violations and eventually reached security at \$82.6M. Episodic memory alone reached security one round earlier on the same scenario but at \$126.4M. This case shows that procedural reuse can guide additional search toward a substantially cheaper secure plan.

Unsuccessful reuse. In `heat_wave_1`, T_{N1} and F_{N1} remained after the first commitment, and procedural guidance was retrieved five times. At the second retrieval, one returned strategy assumed that T_{N1} had already been cleared, although it remained nonzero in the current state. No candidate improved on the retained plan in the following round. SCOPE eventually required six rounds and reached \$69.8M, whereas episodic memory alone required two rounds and reached \$66.9M. This case illustrates a limitation of the current retrieval rule: a historical round can be close in outcome type and Security5 state while the retrieved strategy's trigger is still inappropriate for the current planning state.

G Experimental Settings

Table 7 lists the settings shared by all configurations.

Table 7: Experimental settings.

Setting	Value
Planner and consolidation model	Qwen3.5-9B, served with vLLM, maximum context 32,768 tokens
Planner sampling	temperature 0.3, thinking enabled
Planner output budgets	3,600 tokens per joint A/B/C generation call; 2,400 per plan fusion call, including reasoning
Planning rounds per episode	at most 30
Consolidation sampling	temperature 0.3, thinking enabled
Episodic retrieval	1 entry, for Candidate A only, from other scenarios
Procedural retrieval	up to 2 strategies when retrieval is triggered
Cost refinement	10-round analysis; up to 3 edits per round, temperature 0.3