CAN BROAD BIOMEDICAL KNOWLEDGE BE CONTEXTUALIZED INTO SCENARIO-GROUNDED PROPOSITIONS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Biomedical discovery often requires reconciling broad biomedical knowledge with specific experimental or clinical data. While background knowledge suggests useful biological mechanisms, it is typically too general to map directly onto concrete dataset variables. Conversely, data-driven patterns are often dataset-specific and lack mechanistic insight, creating a gap between abstract principles and concrete evidence. We formulate the gap as *knowledge contextualization*: transforming broad biomedical knowledge into evidence-supported, scenario-grounded propositions, with the goal of allowing domain experts to efficiently inspect, replay, and validate candidate hypotheses. To achieve this, we propose SCENE, a bi-level multi-agent framework that implements knowledge contextualization as an iterative search process. The upper level translates broad knowledge into search directions and maps them onto the dataset schema. The lower level then executes these grounded directions using multi-objective optimization to find concrete propositions that balance evidential strength with sufficient data support. A feedback loop between the two levels progressively refines these search directions. We evaluate SCENE in two distinct settings: discovering patient subgroups with heterogeneous treatment benefits in clinical trial scenarios, and identifying context-specific biological responses in LINCS L1000 studies. In clinical trial scenarios, SCENE outperforms existing baselines by discovering highly specific, strongly supported subgroups. Furthermore, SCENE discovers perturbational contexts with stronger target-response matching and higher strong-response rates in LINCS L1000 studies. These results demonstrate that SCENE effectively bridges the gap between broad knowledge and scenario-specific evidence, providing traceable and inspectable hypotheses for follow-up validation.

1 INTRODUCTION

Biomedical discovery often requires reconciling broad biomedical knowledge with specific experimental or clinical datasets. While background knowledge suggests useful biological mechanisms, it is typically too generic to directly map onto the variables and constraints of a concrete setting. Conversely, the evidence collected in a specific dataset is rich in context but rarely explains how it connects to reusable biomedical principles (Zarin et al., 2011; Lamb et al., 2006; Subramanian et al., 2017). This mismatch creates a practical obstacle: it becomes difficult to prioritize which data patterns deserve follow-up, because statistically strong patterns may lack biological rationale, while plausible biomedical ideas may remain too abstract to test in the available data.

Existing methods address parts of this problem but typically approach it from a single direction, failing to bridge the gap between broad biomedical knowledge and concrete data. On the knowledge side, integration methods inject external knowledge into prompts, representations, or retrieval modules (Wang et al., 2024; Xie et al., 2025). However, this approach leaves the knowledge largely static; it does not specify how broad biomedical knowledge should be grounded under the specific variables, endpoints, and validity constraints of a given dataset. On the evidence side, data-driven algorithms, such as tabular models and subgroup discovery, search for statistical patterns within a single dataset (Foster et al., 2011; Lipkovich et al., 2011; Athey & Imbens, 2016; Athey et al., 2019; Zhou et al., 2024; Yang et al., 2026; Lamb et al., 2006; Subramanian et al., 2017). These discovered

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.

patterns are often weakly tied to prior biomedical knowledge, making them difficult to validate or reuse as traceable candidate hypotheses. Consequently, there remains a lack of an explicit conversion mechanism that translates broad knowledge into data-grounded, scenario-specific propositions.

We therefore formulate this missing conversion step as *knowledge contextualization*: transforming broad biomedical knowledge into evidence-supported, scenario-grounded propositions. The goal is to produce traceable candidate hypotheses that domain experts can efficiently inspect, replay, and validate in follow-up analysis (Chandak et al., 2023; Moreau & Missier, 2013). This problem is challenging for three reasons. First, broad biomedical concepts do not directly match the concrete variables and measurements in a specific dataset. For example, a concept such as *metabolic dysregulation* may need to be instantiated using measurements such as BMI, fasting glucose, or triglycerides, depending on the dataset schema. Second, the same biomedical principle requires different grounding across distinct settings; for example, it may manifest as a patient subgroup in a clinical trial, or as a context-specific cellular response in a perturbational assay (Foster et al., 2011; Subramanian et al., 2017). Third, even after mapping these concepts to concrete variables, the space of valid candidate propositions is immense, making exhaustive search computationally impractical (Lipkovich et al., 2024; Zhou et al., 2024). Addressing these challenges requires an explicit contextualization process that can ground biomedical priors, evaluate them against available evidence, and return interpretable results; Figure 1 contrasts rule-centered discovery with evidence-driven knowledge contextualization.

To address these challenges, we propose SCENE, a bi-level multi-agent framework that implements knowledge contextualization as an iterative search process (Hong et al., 2024; Wu et al., 2024; Zhang et al., 2025a). At the upper level, SCENE translates broad biomedical knowledge into testable search directions and formulates a schema-grounding plan. At the lower level, it executes this plan using multi-objective optimization to find concrete propositions that balance evidential strength with sufficient data support (Deb et al., 2002). Crucially, the two levels form a bidirectional closed loop: the upper level constrains the search space based on biomedical priors, while the scenario evidence from the lower level guides the upper level to refine or pivot the search directions. By explicitly linking knowledge-guided generation with data-driven evaluation, SCENE provides a traceable mechanism to turn abstract knowledge into scenario-grounded propositions.

- We formulate *knowledge contextualization* as a standalone problem in biomedical discovery: transforming broad biomedical knowledge into evidence-supported, scenario-grounded propositions to produce traceable, inspectable candidate hypotheses.
- We propose SCENE, a bi-level multi-agent framework that forms a bidirectional closed loop between upper-level direction planning and lower-level multi-objective evolutionary search: the upper level guides the lower level by constraining and prioritizing its executable search space, while the lower level feeds scenario-specific evidence back to the upper level to refine subsequent directions.
- We demonstrate SCENE's generality across two distinct biomedical settings: clinical-trial treatment-benefit subgroup discovery and LINCS L1000 perturbational context discovery. In clinical trials, SCENE outperforms existing baselines by discovering subgroups with higher risk reduction and stronger data support. In L1000 studies, it discovers perturbational contexts with stronger target-response matching and higher strong-response rates than evaluated baselines. We further show that SCENE propositions can improve downstream few-shot classification.

2 Related Work

Knowledge integration in biomedicine. A line of work incorporates external biomedical knowledge into machine learning systems through knowledge graphs, structured retrieval, or knowledge injection. Recent efforts such as HypKG (Xie et al., 2025) contextualize biomedical knowledge graphs with patient-specific EHR information, while benchmarks such as FEDMEKI (Wang et al., 2024) study how medical knowledge can be injected into foundation models under privacy constraints. However, these approaches primarily operate at the level of representation learning or model training, aiming to improve downstream prediction or generation, rather than directly producing interpretable, scenario-grounded propositions in the current setting.

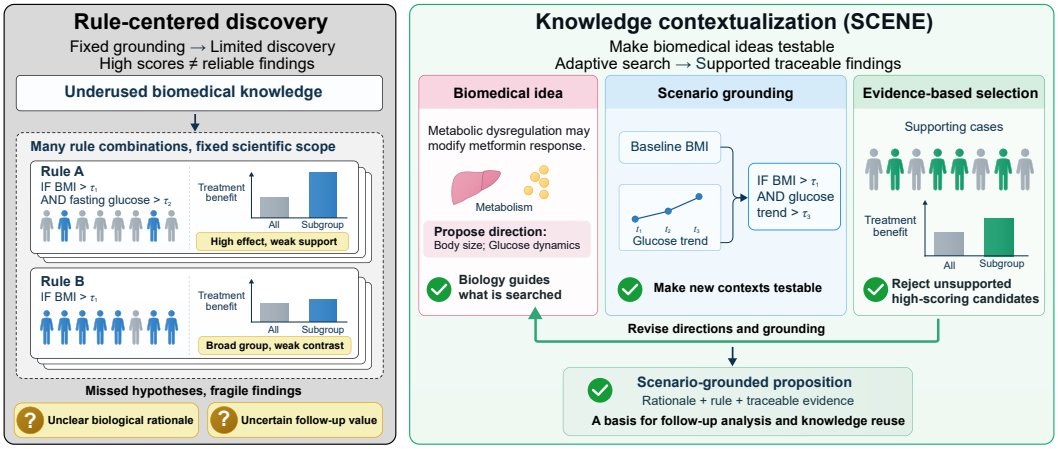


Figure 1: From rule search to knowledge contextualization. Fixed grounding can leave relevant hypotheses unexplored, while high-scoring rules may lack adequate support. SCENE makes biomedical ideas testable, revises search directions and grounding using discovery evidence, and balances effect with support. Its scenario-grounded propositions link biomedical rationale, executable rules, and traceable evidence for follow-up validation and knowledge reuse. The clinical example is schematic.

Single-scenario subgroup discovery and treatment effect heterogeneity. Another related literature studies treatment effect heterogeneity and subgroup discovery, especially in clinical trials. Classical methods such as Virtual Twins (Foster et al., 2011) and SIDES (Lipkovich et al., 2011) search for treatment-responsive subgroups through predicted individual responses or recursive partitioning, while honest trees (Athey & Imbens, 2016) and causal forests (Athey et al., 2019) extend this line through tree- or forest-based estimation of heterogeneous treatment effects. More recent approaches, such as CURLS (Zhou et al., 2024) and Learning Subgroups with Maximum Treatment Effects without Causal Heuristics (Yang et al., 2026) cast subgroup discovery as rule learning or structured optimization. Despite their differences, these methods share a common perspective: they treat the problem as data-only effect estimation, partitioning, or rule search within a given dataset. They neither use broad biomedical knowledge as an explicit hypothesis space, nor address how such knowledge can be converted into scenario-grounded propositions.

Our position. SCENE is not defined around subgroup discovery, treatment effect estimation, or any single downstream task. Instead, we study *knowledge contextualization* itself: how broad biomedical knowledge can be proposed, grounded, tested, and refined in a concrete scenario. Under this view, subgroup rules in clinical trials are only one form of scenario-grounded proposition, alongside context-bounded findings in perturbational transcriptomic studies. SCENE uses scenario evidence to revise how knowledge is grounded and what is searched next, rather than only optimizing rules within a fixed feature set.

3 Methodology

3.1 Task and Overview

For a biomedical scenario s, SCENE receives three inputs: biomedical prior knowledge $\mathcal{K}^{(s)}$, scenario evidence $\mathcal{D}^{(s)} = (X^{(s)}, Y^{(s)}, C^{(s)})$, and a machine-readable schema $\mathcal{S}^{(s)}$. Here $X^{(s)}$ denotes observable features, $Y^{(s)}$ denotes outcomes or response variables used for evidence evaluation, and $C^{(s)}$ denotes task context. The schema $\mathcal{S}^{(s)}$ specifies observable variables, admissible windows or operators, forbidden fields for candidate construction, and validity constraints. Biomedical prior knowledge $\mathcal{K}^{(s)}$ is treated as a finite set of pre-discovery records, such as statements derived from publications, trial records, or study resources. These records provide source-traceable starting points for search directions. Detailed record formats, manifest fields, and provenance requirements are given in Appendices A.1 and A.7.

The target output is a set of scenario-grounded propositions

$$\mathcal{P}^{(s)} = \{p_i\}_{i=1}^{m}, \qquad p_i = (d_i, q_i, \omega_i). \tag{1}$$

Here d_i is a search direction specifying the biomedical concept or relation to explore, q_i is an executable rule or finding grounded in the current scenario, and ω_i is an evidence record containing effect, support, diagnostics, source links, and provenance. The grounded object q_i may be a subgroup rule in clinical discovery or a cell–perturbagen–dose–time context in L1000. Its evidence record makes the proposition inspectable and replayable. Thus, the reportable unit is the entire proposition (d_i, q_i, ω_i), not a rule or finding alone.

SCENE implements knowledge contextualization as a closed-loop search process. The upper level translates prior knowledge and task context into bounded search directions and schema-grounding plans, which constrain and prioritize the executable search space. The lower level searches admissible candidate rules or findings with multi-objective optimization over effect and support, and returns scenario-specific feedback for later direction refinement. The final outputs are proposition-level artifacts that combine a direction, an executable grounded object, and an evidence record.

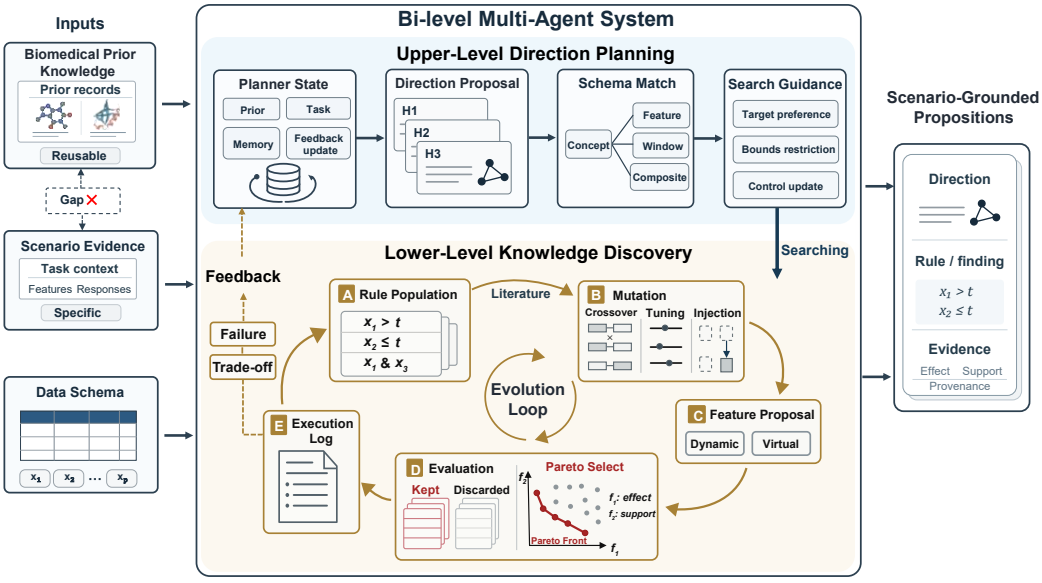


Figure 2: SCENE (Scenario-Contextualized Evidence and Knowledge Engine) artifact flow. Biomedical prior knowledge, scenario evidence, and a data schema define the discovery scenario. The upper level converts broad knowledge into bounded search directions and schema-grounding plans that constrain and prioritize the lower-level executable search space. The lower level searches candidate rules or findings with multi-objective effect/support optimization and returns scenario-specific feedback to refine later directions. Exported outputs are scenario-grounded propositions, each pairing a direction, an executable rule or finding, and an evidence record with provenance.

Figure 2 summarizes the operational flow. Given an active direction at round t, the schema-grounding plan $\mathcal{G}_t$ induces an admissible candidate family $\mathcal{Q}_t$. A scenario adapter $\mathcal{A}^{(s)}$ specifies the executable candidate language, validity checks, support and effect functions, and reporting boundary. Details of the candidate grammar, grounding operators, reporting checks, and scenario-specific estimators are deferred to Appendices A.6, A.7, and A.8. The round-level protocol is shown in Algorithm 1.

3.2 Upper-Level Direction Planning

The upper level decides what should be searched next. It receives prior records, task context, schema summaries, and discovery-visible feedback from previous rounds. Its output is not a biomedical claim, but a bounded search direction together with search guidance:

$$(d_t, \eta_t) = \mathrm{Plan}_\theta(\mathcal{K}^{(s)}, C^{(s)}, \mathcal{S}^{(s)}, m_{t-1}, f_{t-1}, \eta_t^0), \qquad \eta_t = (\pi_t, H_t, \rho_t, \beta_t). \tag{2}$$

Algorithm 1 SCENE round-level search protocol

Require: prior records $\mathcal{K}^{(s)}$, scenario evidence $\mathcal{D}^{(s)}$, schema $\mathcal{S}^{(s)}$, adapter $\mathcal{A}^{(s)}$, round budget T
Ensure: reportable propositions $\mathcal{P}^{(s)}$

initialize compact state m_0, feedback f_0, controls η_1^0, and $\mathcal{P}^{(s)} \leftarrow \emptyset$
for $t \leftarrow 1$ to T **do**
  $(d_t, \eta_t) \leftarrow \textsc{Plan}(\mathcal{K}^{(s)}, C^{(s)}, \mathcal{S}^{(s)}, m_{t-1}, f_{t-1}, \eta_t^0)$ ▷ propose a bounded search direction
  $\mathcal{G}_t \leftarrow \textsc{Ground}(d_t, \mathcal{S}^{(s)}, \eta_t)$ ▷ map the direction to schema-visible objects
  $(\mathcal{G}_t, \eta_t) \leftarrow \textsc{Validate}(\mathcal{G}_t, \eta_t, \mathcal{A}^{(s)})$ ▷ remove invalid or leakage-risk objects
  $\mathcal{C}_{t,0} \leftarrow \textsc{Seed}(\mathcal{G}_t, \eta_t)$ ▷ initialize candidate rules or findings
  $(\mathcal{F}_t, \lambda_t) \leftarrow \textsc{Evolve}(d_t, \mathcal{G}_t, \mathcal{C}_{t,0}, \mathcal{D}^{(s)}, \mathcal{A}^{(s)}, \eta_t)$ ▷ search and retain a Pareto frontier
  $(f_t, m_t, \bar{\eta}_{t+1}) \leftarrow \textsc{UpdateFeedback}(\mathcal{F}_t, \lambda_t, d_t, m_{t-1})$ ▷ update next-round search using discovery-visible logs
  $\mathcal{P}^{(s)} \leftarrow \mathcal{P}^{(s)} \cup \textsc{Report}(\mathcal{F}_t, d_t, \mathcal{D}^{(s)}, \mathcal{A}^{(s)})$ ▷ export propositions passing reporting checks
  $\eta_{t+1}^0 \leftarrow \bar{\eta}_{t+1}$
end for
return $\mathcal{P}^{(s)}$

Here π_t is the operator mixture, H_t is the generation budget, ρ_t is the support floor, and β_t contains bounded preferences such as target feature families, admissible ranges, avoid lists, or virtual/dynamic feature toggles. The symbol θ denotes fixed role instructions, decoding settings, controller bounds, and replay policy, all specified before the discovery run.

Operationally, PLAN performs three functions. First, it selects or revises a search direction from prior knowledge, task context, and prior feedback. Second, it grounds the direction to schema-visible feature families, windows, or admissible composites. Third, it converts the grounded direction into search guidance for the lower level. All role outputs, including directions and grounding plans, are treated as structured proposals and must pass deterministic validation before entering executable search. Detailed role contracts and controller policies are given in Appendices A.2 and A.3.

3.3 LOWER-LEVEL KNOWLEDGE DISCOVERY

Conditioned on the active direction and its grounding plan, the lower level searches over an admissible candidate family $\mathcal{Q}_t$. Candidates are executable scenario-specific objects: subgroup rules in clinical settings and context-bounded findings in L1000 settings. The search starts from grounded seeds and diversity-oriented admissible candidates, then applies variation operators such as mutation, crossover, tuning, and injection. When raw observables are insufficient to instantiate a direction, virtual or dynamic features may be proposed, but they enter the search space only after deterministic construction, provenance recording, and validity checks.

Each candidate $q \in \mathcal{Q}_t$ is evaluated by the scenario adapter, which returns an effect objective $J_{\text{eff}}^{(s)}(q)$, a support objective $J_{\text{sup}}^{(s)}(q)$, and discovery-visible diagnostics $V^{(s)}(q)$. The effect objective measures the configured scenario contrast, such as a treatment-benefit contrast in clinical subgroup discovery or a target-response contrast in L1000 discovery. The support objective prevents the search from favoring highly specific but poorly supported fragments. These objectives define discovery-side ranking under the frozen split and manifest. Pareto dominance and frontier membership use only effect and support:

$$\mathcal{F}_t = \left\{ q \in \mathcal{Q}_t : \nexists q' \in \mathcal{Q}_t \text{ such that } \begin{array}{l} J_{\text{eff}}^{(s)}(q') \geq J_{\text{eff}}^{(s)}(q), \\ J_{\text{sup}}^{(s)}(q') \geq J_{\text{sup}}^{(s)}(q), \\ \text{and at least one inequality is strict} \end{array} \right\}. \tag{3}$$

The retained frontier is both a candidate set and a feedback object. A coherent frontier indicates that the current direction admits repeated grounded instantiations, whereas empty, low-support, invalid, or redundant frontiers expose failure modes. Diagnostics can annotate instability or break exact ties after Pareto ranking and diversity filtering, but they cannot create or remove Pareto dominance. Evolutionary operators and default hyperparameters are detailed in Appendix A.5; executable grounding operators are described in Appendix A.6.

3.4 Closed-Loop Interaction and Reporting Boundary

The upper and lower levels interact through validated artifacts and frontier logs rather than unconstrained dialogue. At each round, the upper level sends a direction, a grounding plan, and bounded search guidance to shape the lower-level search; the lower level returns a retained frontier $\mathcal{F}_t$ and an execution log λ_t for the next planning step. The feedback update summarizes failure modes, support–effect trade-offs, alignment gaps, diversity, and diagnostic contrast:

$$(f_t, m_t, \bar{\eta}_{t+1}) = \Phi_{\text{fb}}(\mathcal{F}_t, \lambda_t, d_t, m_{t-1}). \tag{4}$$

These signals help later rounds preserve, narrow, broaden, or replace search directions.

Export is stricter than feedback. A candidate may influence later search if it is discovery-visible, but it becomes a reported proposition only if it lies on the retained frontier and passes validity, support, provenance, leakage, and diagnostic checks. Final held-out or test evidence is used only for post-discovery audit and is never fed back into the same run to revise directions, rules, findings, or search controls. SCENE therefore links validated knowledge grounding and effect/support search with evidence-driven revision, while preserving the direction, executable finding, and provenance in each reported proposition. Detailed split-lineage and reporting schemas are given in Appendix A.7.

4 Experiments

4.1 Experimental Setup

Table 1: Held-out clinical subgroup discovery. Frozen Best-1 rules are replayed over 50 paired splits. Task columns report held-out bad-outcome ARR, B/P denote breast-cancer/PCOS tasks and BL/TR denote baseline/trajectory views. P-ARR is population-adjusted ARR, U-Supp. is usable support, and Best-1 DC is train-to-holdout direction consistency. Higher is better for all displayed metrics.

Method	B-D-BL	B-D-TR	P-M-BL	P-M-TR	P-A-FPG	P-A-AUC	P-ARR ↑	U-Supp. ↑	Best-1 DC ↑
Virtual Twins	0.023	-0.027	0.036	0.197	0.078	0.004	0.014	58.7	51.5
SIDES	0.007	0.008	0.195	-0.005	0.006	-0.058	0.010	26.7	74.3
Causal Forest	0.066	0.005	0.104	0.035	0.060	0.130	0.016	56.3	51.4
Honest Causal Tree	0.057	0.037	0.232	0.042	0.053	0.087	0.032	70.0	71.7
RuAG	0.004	0.002	0.117	0.113	0.127	0.008	0.016	33.3	49.0
CURLS	0.015	0.003	0.025	0.210	0.118	0.103	0.010	26.3	45.7
MaxTE Subgroups	0.076	0.012	0.234	0.121	0.117	0.132	0.054	38.3	68.6
SCENE (ours)	**0.086**	**0.095**	**0.441**	**0.439**	**0.269**	**0.270**	**0.093**	**84.7**	**100.0**

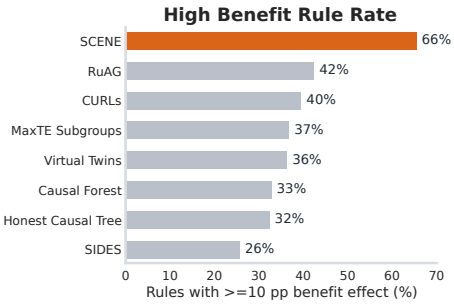


Figure 3: Fraction of exported rules with treatment-benefit effect at least 10 percentage points.

Method	Conn. ↑	Strong-Pos. ↑	AUPRC ↑	U-Supp. ↑	Gap ↓
MaxTE-Context	0.104	18.5	**0.334**	**78.8**	0.228
One-shot LLM	0.107	24.4	0.253	38.3	0.129
Minimal	0.166	27.6	0.260	33.5	0.111
w/o Dynamic	0.171	28.5	0.273	34.2	0.068
w/o Pareto	0.184	18.8	0.273	30.0	0.081
w/o Upper	0.177	35.1	0.271	29.5	0.106
w/o Virtual	0.170	24.2	0.275	34.3	0.087
SCENE (ours)	**0.188**	**41.5**	0.268	31.4	**0.063**

Table 2: L1000 baselines and component ablations. Conn. is connectivity margin, Strong-Pos. the strong-response rate, U-Supp. the usable-rule fraction (%), and Gap the normalized train–holdout gap.

We evaluate SCENE on two discovery settings: clinical subgroup discovery and perturbational response discovery, and one downstream knowledge-reuse setting. The clinical benchmark uses patient-level tables linked to two registered trials, with NCT identifiers used only for registry and protocol linkage: NCT00174655 from Project Data Sphere and NCT02491333 from the Dryad

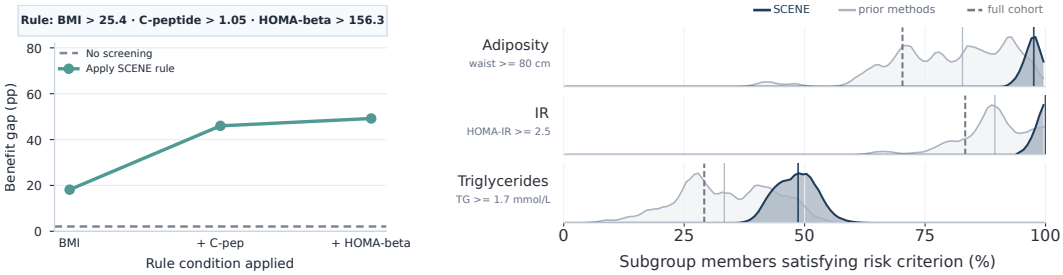


Figure 4: Concrete clinical meaning of a SCENE subgroup rule. (Left) The displayed SCENE rule increases the favorable outcome treatment–control gap. (Right) Rule-level clinical-risk audit. Gray densities summarize valid top-ranked rules from prior methods replayed on the same cohort; blue densities summarize sensitivity-only local threshold perturbations around the SCENE rule.

release (Project Data Sphere, 2026; Green et al., 2015; Wen et al., 2021; 2022). We selected the trials by feasibility and coverage criteria: randomized arms, patient-level covariates, interpretable endpoints, sufficient support for repeated subgroup replay, and complementary biomedical settings. We compare against subgroup and heterogeneous treatment-effect baselines (Foster et al., 2011; Lipkovich et al., 2011; Athey & Imbens, 2016; Athey et al., 2019; Zhou et al., 2024; Yang et al., 2026). We adapt *RuAG* (Zhang et al., 2025b) as an LLM-assisted rule-search baseline, combining LLM-based feature selection with Monte Carlo tree search for clinical subgroups. Across 50 paired splits, each run searches the training split, exports one prioritized rule, and replays it on held-out patients for six task frames: B-D-BL, B-D-TR, P-M-BL, P-M-TR, P-A-FPG, P-A-AUC (Appendix A.10). The primary clinical metric is held-out bad-outcome ARR (absolute risk reduction, larger is better). P-ARR records population-adjusted benefit, U-Supp. records usable support, and Best-1 DC records train-to-holdout benefit-direction preservation over committed Best-1 rules, definitions are in Appendix A.11. In the perturbational benchmark, based on CMap/LINCS L1000 perturbational signatures (Lamb et al., 2006; Subramanian et al., 2017; Keenan et al., 2018; Koleti et al., 2018), each episode fixes a target mechanism and performs discovery. We report Conn. as the held-out connectivity margin between rule-covered signatures and background, Strong-Pos. as the strong response rate, AUPRC (Davis & Goadrich, 2006; Saito & Rehmsmeier, 2015) as target mechanism retrieval, U-Supp. as usable held-out support, and Gap as the normalized train–holdout connectivity gap. *MaxTE-Context* adapts MaxTE (Yang et al., 2026) to mine L1000 context rules without biomedical priors; *One-shot LLM* propose rules from priors and the schema without iterative refinement; Appendix A.12 gives the replay contract. Finally, the reuse setting compares downstream few-shot classification with C0 (few-shot row serialization), C1 (C0 plus generic knowledge context), and C2 (C1 plus SCENE knowledge cards), using final-test AUPRC, MacroF1 (Sokolova & Lapalme, 2009), and accuracy to measure ranking, class-balanced F1, and correctness. Held-out outcomes and connectivity are never used to select rules or revise prompts, downstream test labels are never used to choose SCENE cards.

4.2 Clinical Trial Subgroup Discovery

SCENE has the highest held-out ARR point estimate in all six frames (Table 1): on baseline static frames (B-D-BL, P-M-BL), it outperforms existing SOTA methods (0.086 vs. 0.076; 0.441 vs. 0.234) and retains its lead on trajectory-enhanced dynamic frames (B-D-TR, P-M-TR, P-A-FPG, P-A-AUC), indicating that the advantage is not tied to one endpoint or table view. SCENE also outperforms existing SOTA methods in aggregate diagnostics : with P-ARR of 0.093 and usable support of 84.7%, versus baseline maxima of 0.054 and 70.0%. Compared with the LLM-assisted *RuAG* baseline, SCENE achieves higher ARR on all six frames, alongside higher P-ARR (0.093 vs. 0.016), usable support (84.7% vs. 33.3%), and direction consistency (100.0% vs. 49.0%). Thus, SCENE handles diverse clinical tables while discovering support-aware propositions.

4.3 L1000 Perturbational Context Discovery

Table 2 evaluates SCENE in the L1000 mechanism discovery setting by comparing *MaxTE-Context*, *One-shot LLM* and using component ablations to examine how different modules affect perturbational context discovery. SCENE achieves stronger connectivity and response concentration with a smaller

train–holdout gap than both controls. Removing Pareto selection (w/o Pareto) primarily collapses the high-response tail. Removing upper-level planning (w/o Upper) produces a different failure mode: the strong positive rate falls and the gap increases, suggesting weaker transfer from discovery to the holdout set. Furthermore, removing virtual features (w/o Virtual) trades effect strength for coverage. These results indicate that the components are complementary rather than redundant. Specifically, upper-level direction planning stabilizes discovery across the boundary between training and holdout sets, Pareto selection preserves strong holdout responders, and virtual or dynamic grounding expands the search space without replacing the need for evidence-based frontier selection.

4.4 Contextual Knowledge Reuse

Table 3 evaluates whether propositions discovered by SCENE can be reused as auxiliary knowledge for downstream few-shot tabular classification. We convert exported discovery propositions into a fixed pool of compact knowledge cards, select cards using downstream validation labels, and inject them into in-context prediction. Row serialization, exemplar budget, data split, and LLM decoding remain fixed across conditions. The comparison includes ordinary few-shot row serialization (C0), C0 plus generic scene context (C1), and C1 plus row-conditioned SCENE knowledge cards (C2). In the clinical scene, the downstream label is a row-level proxy for treatment-benefit evidence under the task endpoint; in the L1000 scene, the downstream label is the target-mechanism positive class defined for the selected perturbational task. C2 is best on all final test metrics in both scenarios, improving clinical accuracy from 0.5510 to 0.6939 and L1000 accuracy from 0.6100 to 0.7200 over C1. These results suggest that SCENE propositions provide reusable scenario-conditioned information beyond generic task descriptions, while serving as downstream utility evidence rather than independent biomedical validation. Seed-level summaries and prompt artifacts are audited in Appendix A.13.

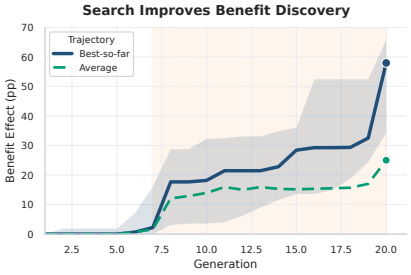


Figure 5: Discovery-side SCENE search trajectory showing median best-so-far and mean candidate benefit across generations.

Scenario	Mode	AUPRC	MacroF1	ACC
Clinical Trial	C0	0.5877	0.4408	0.4898
	C1	0.6140	0.4745	0.5510
	C2	**0.6747**	**0.5291**	**0.6939**
L1000	C0	0.3582	0.5536	0.5600
	C1	0.3725	0.5914	0.6100
	C2	**0.4113**	**0.6528**	**0.7200**

Table 3: Few-shot reuse of SCENE propositions. C0: row-serialized few-shot baseline; C1: C0 + generic knowledge context; C2: C1 + validation-selected, row-conditioned SCENE cards. Metrics are means over 10 exemplar seeds.

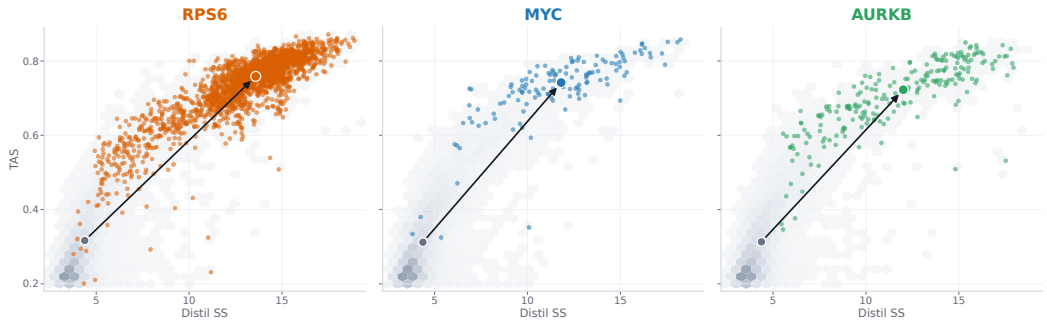
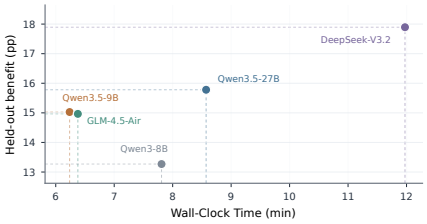


Figure 6: Held-out L1000 rule replay on 3 target mechanisms. Gray hexbins summarize the held-out background, colored points denote signatures satisfying the selected rule, and arrows indicate the median shift from the background to rule-covered signatures.

4.5 Search, Replay, and Efficiency Diagnostics

To complement the aggregate results, we further examine representative propositions, replay behavior, and efficiency diagnostics. Figure 4 shows that a metabolic P-A-AUC rule raises the favorable treatment–control gap and remains enriched for adiposity, insulin resistance, and elevated triglycerides relative to the rule distributions induced by prior methods in Table 1, suggesting a stable, task-relevant endocrine–metabolic phenotype rather than a clinically unstructured output. Figure 6 replays exported rules for RPS6, MYC, and AURKB. Profiles consistently shift toward higher Distil SS and TAS (L1000 signature-strength and transcriptional-activity diagnostics (Subramanian et al., 2017)), indicating that SCENE identifies coherent response contexts rather than isolated score gains. Figure 3 shows that 66% of SCENE's reported clinical rules exceed a 10-percentage-point benefit threshold. Figure 5 shows that both the median best-so-far and the average discovery-side benefit among candidates improve across generations. Together with Table 3, these analyses support the view that SCENE outputs compact, scenario-grounded propositions that remain reusable outside the original discovery run. Figure 7 compares different API backends on the P-M-TR task, Qwen3.5-9B provides the most favorable runtime–benefit trade-off.



Backend	Avg. run (min)	Avg. gen. (s)
Qwen3.5-9B (Qwen Team, 2026c)	**6.2**	**19.8**
GLM-4.5-Air (Z.AI, 2025)	6.4	26.9
Qwen3-8B (Qwen Team, 2025)	7.8	25.8
Qwen3.5-27B (Qwen Team, 2026b)	8.6	31.2
DeepSeek-V3.2 (DeepSeek-AI, 2025)	12.0	43.7

Figure 7: Benefit–runtime trade-off and runtime cost across LLM backends. (Left) Each point is one third-party API backend under the same discovery protocol. (Right) Avg. run is mean wall-clock time per discovery run; Avg. gen. is mean wall-clock time per evolutionary generation within a run.

4.6 Further Analysis

Appendix B provides post-hoc audits beyond aggregate benchmark scores. Appendix B.1 analyzes clinical benefit–support–variability geometry, showing that SCENE remains high-benefit, high-support, and low-variability. Appendix B.2 compares reported rules on predefined clinical axes, showing stronger alignment with task-relevant phenotypes than prior-method rule. Appendix B.3 audits L1000 inspectability, showing improved response-quality diagnostics. Appendix B.4 evaluates replay reliability, showing that benefit directions are largely preserved and magnitude stability improves. Appendix B.5 analyzes rule composition and provenance, showing that SCENE combines interpretable feature families rather than opaque rule strings. Appendix B.6 expands the L1000 ablation, showing complementary contributions from different components. Appendix B.7 shows that generated features are validated before entering reported propositions. Appendix B.8 audits grounding traces, showing a visible path from broad knowledge to executable propositions. It also examines planner consistency and report quality. Appendix B.9 reports knowledge-guided controls and prior sensitivity. Together, these analyses support the view that SCENE outputs robust, inspectable, and provenance-aware scenario-grounded propositions rather than isolated high-scoring rules.

5 Conclusion

In this work, we studied knowledge contextualization as a biomedical discovery problem: converting broad prior knowledge into evidence-supported, scenario-grounded propositions. We proposed SCENE, a bi-level multi-agent framework that couples direction planning, schema grounding, and multi-objective evidence search to produce traceable propositions rather than isolated data patterns. Compared with the evaluated baselines, SCENE achieves higher held-out ARR and aggregate usable-support rates in clinical subgroup discovery, higher connectivity margins and strong-response rates in LINCS L1000, and further improves downstream few-shot classification. These results suggest that broad biomedical knowledge can be operationalized as auditable, scenario-grounded hypotheses for follow-up validation.

AI USE STATEMENT

Generative AI tools were used to improve the presentation of author-written text, including grammar, phrasing, clarity, and readability. This writing assistance was limited to language polishing and did not alter the scientific content of the manuscript. Generative AI tools also assisted with literature search and the identification of relevant prior work. LLM use within SCENE and the experiments is described in Section 3 and Appendix A, with two AI evaluators used for the exploratory report-quality assessment in Appendix B.8. All AI-assisted edits were reviewed and, where appropriate, revised by the authors.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide the concrete settings used in all reported experiments, including the scenario-specific data schemas, LLM role configurations, multi-objective evolutionary hyperparameters, threshold grids, and validation split boundaries. **The source code and processed datasets are included in the supplemental material submitted with this paper.** We will also release the code publicly upon acceptance. In addition, we give explicit definitions of the operational boundaries used by SCENE, such as the role instruction contracts, deterministic reporting predicates, and scenario-specific estimators, so that the bi-level knowledge contextualization procedure can be reproduced as faithfully as possible.

REFERENCES

Susan Athey and Guido Imbens. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27):7353–7360, 2016. doi: 10.1073/pnas.1510489113.

Susan Athey, Julie Tibshirani, and Stefan Wager. Generalized random forests. *The Annals of Statistics*, 47(2):1148–1178, 2019. doi: 10.1214/18-AOS1709.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of the Conference on Neural Information Processing Systems*, pp. 1877–1901, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Payal Chandak, Kexin Huang, and Marinka Zitnik. Building a knowledge graph to enable precision medicine. *Scientific Data*, 10(1):67, 2023. doi: 10.1038/s41597-023-01960-3. URL https://doi.org/10.1038/s41597-023-01960-3.

ClinicalTrials.gov. ClinicalTrials.gov record for NCT00174655. https://clinicaltrials.gov/study/NCT00174655, 2011. ClinicalTrials.gov identifier NCT00174655; accessed 2026-04-28.

ClinicalTrials.gov. ClinicalTrials.gov record for NCT02491333. https://clinicaltrials.gov/study/NCT02491333, 2022. ClinicalTrials.gov identifier NCT02491333; accessed 2026-04-28.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and ROC curves. In *Proceedings of the International Conference on Machine Learning*, pp. 233–240, 2006. doi: 10.1145/1143844.1143874.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. doi: 10.1109/4235.996017.

DeepSeek-AI. DeepSeek-V3.2 release. https://api-docs.deepseek.com/news/news251201, 2025. Accessed 2026-05-07.

Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. doi: 10.1016/j.patrec.2005.10.010.

Jared C. Foster, Jeremy M. G. Taylor, and Stephen J. Ruberg. Subgroup identification from randomized clinical trial data. *Statistics in Medicine*, 30(24):2867–2880, 2011. doi: 10.1002/sim.4322.

Angela K. Green, Katherine E. Reeder-Hayes, Robert W. Corty, Ethan Basch, Mathew I. Milowsky, Stacie B. Dusetzina, Antonia V. Bennett, and William A. Wood. The project data sphere initiative: Accelerating cancer research by sharing data. *The Oncologist*, 20(5):464–e20, 2015. doi: 10.1634/theoncologist.2014-0431.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *Proceedings of the International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VtmBAGCN7o.

Alexandra B. Keenan, Sherry L. Jenkins, Kathleen M. Jagodnik, Simon Koplev, Edward He, Denis Torre, Zichen Wang, Anders B. Dohlman, Moshe C. Silverstein, Alexander Lachmann, Maxim V. Kuleshov, Avi Ma'ayan, Vasileios Stathias, Raymond Terryn, Daniel Cooper, Michele Forlin, Amar Koleti, Dusica Vidovic, Caty Chung, Stephan C. Schürer, et al. The library of integrated network-based cellular signatures NIH program: System-level cataloging of human cells response to perturbations. *Cell Systems*, 6(1):13–24, 2018. doi: 10.1016/j.cels.2017.11.001.

David M. Kent, Jessica K. Paulus, David van Klaveren, Ralph D'Agostino, Steven Goodman, Rodney Hayward, John P. A. Ioannidis, Bray Patrick-Lake, Sally Morton, Michael Pencina, Gowri Raman, Joseph S. Ross, Harry P. Selker, Ravi Varadhan, Andrew Vickers, John B. Wong, and Ewout W. Steyerberg. The predictive approaches to treatment effect heterogeneity (PATH) statement. *Annals of Internal Medicine*, 172(1):35–45, 2020. doi: 10.7326/M18-3667.

Amar Koleti, Raymond Terryn, Vasileios Stathias, Caty Chung, Daniel J. Cooper, John P. Turner, Dusica Vidovic, Michele Forlin, Tanya T. Kelley, Alessandro D'Urso, et al. Data portal for the library of integrated network-based cellular signatures (LINCS) program: Integrated access to diverse large-scale cellular perturbation response data. *Nucleic Acids Research*, 46(D1):D558–D566, 2018. doi: 10.1093/nar/gkx1063.

Justin Lamb, Emily D. Crawford, David Peck, Joshua W. Modell, Irene C. Blat, Matthew J. Wrobel, Jim Lerner, Jean-Philippe Brunet, Aravind Subramanian, Kenneth N. Ross, Michael Reich, Haley Hieronymus, Guo Wei, Scott A. Armstrong, Stephen J. Haggarty, Paul A. Clemons, Ru Wei, Steven A. Carr, Eric S. Lander, and Todd R. Golub. The connectivity map: Using gene-expression signatures to connect small molecules, genes, and disease. *Science*, 313(5795):1929–1935, 2006. doi: 10.1126/science.1132939.

Arthur Liberzon, Chet Birger, Helga Thorvaldsdóttir, Mahmoud Ghandi, Jill P. Mesirov, and Pablo Tamayo. The molecular signatures database hallmark gene set collection. *Cell Systems*, 1(6):417–425, 2015. doi: 10.1016/j.cels.2015.12.004.

Ilya Lipkovich, Alex Dmitrienko, Jonathan Denne, and Gregory Enas. Subgroup identification based on differential effect search—a recursive partitioning method for establishing response to treatment in patient subpopulations. *Statistics in Medicine*, 30(21):2601–2621, 2011. doi: 10.1002/sim.4289.

Ilya Lipkovich, David Svensson, Bohdana Ratitch, and Alex Dmitrienko. Modern approaches for evaluating treatment effect heterogeneity from clinical trials and observational data. *Statistics in Medicine*, 43(22):4388–4436, 2024. doi: 10.1002/sim.10167. URL https://doi.org/10.1002/sim.10167.

Luc Moreau and Paolo Missier. PROV-DM: The PROV data model. W3C Recommendation REC-prov-dm-20130430, World Wide Web Consortium, April 2013. URL https://www.w3.org/TR/2013/REC-prov-dm-20130430/. Edited by Luc Moreau and Paolo Missier.

Project Data Sphere. Project Data Sphere: Data access. https://data.projectdatasphere.org/projectdatasphere/html/access, 2026. Accessed 2026-05-05.

Qwen Team. Qwen3: Think deeper, act faster. https://qwenlm.github.io/blog/qwen3/, 2025. Accessed 2026-05-07.

Qwen Team. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5, 2026a. Accessed 2026-05-03.

Qwen Team. Qwen/Qwen3.5-27B model card. https://huggingface.co/Qwen/Qwen3.5-27B, 2026b. Accessed 2026-05-07.

Qwen Team. Qwen/Qwen3.5-9B model card. https://huggingface.co/Qwen/Qwen3.5-9B, 2026c. Accessed 2026-05-07.

Peter M. Rothwell. Subgroup analysis in randomised controlled trials: Importance, indications, and interpretation. *The Lancet*, 365(9454):176–186, 2005. doi: 10.1016/S0140-6736(05)17709-5.

Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3):e0118432, 2015. doi: 10.1371/journal.pone.0118432. URL https://doi.org/10.1371/journal.pone.0118432.

Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009. doi: 10.1016/j.ipm.2009.03.002.

Aravind Subramanian, Rajiv Narayan, Steven M. Corsello, David D. Peck, Ted E. Natoli, Xiaodong Lu, Joshua Gould, John F. Davis, Andrew A. Tubelli, Jacob K. Asiedu, David L. Lahr, Jodi E. Hirschman, Zihan Liu, Melanie Donahue, Bina Julian, Mariya Khan, David Wadden, Ian C. Smith, Daniel Lam, Arthur Liberzon, Courtney Toder, Mukta Bagul, Marek Orzechowski, Oana M. Enache, Federica Piccioni, Sarah A. Johnson, Nicholas J. Lyons, Alice H. Berger, Alykhan F. Shamji, Angela N. Brooks, Anita Vrcic, Corey Flynn, Julien Rosains, David Y. Takeda, Roger Hu, David Davison, Justin Lamb, Kristin Ardlie, Larson Hogstrom, Peyton Greenside, Nathanael S. Gray, Paul A. Clemons, Serena Silver, Xiaoyun Wu, Wen-Ning Zhao, Wilfried Read-Button, Xiaohua Wu, Stephen J. Haggarty, Lucienne V. Ronco, Jesse S. Boehm, Stuart L. Schreiber, John G. Doench, Joshua A. Bittker, David E. Root, Bang Wong, and Todd R. Golub. A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell*, 171(6):1437–1452.e17, 2017. doi: 10.1016/j.cell.2017.10.049.

Jiaqi Wang, Xiaochen Wang, Lingjuan Lyu, Jinghui Chen, and Fenglong Ma. FEDMEKI: A benchmark for scaling medical foundation models via federated knowledge injection. In *Proceedings of the Conference on Neural Information Processing Systems*, 2024. Datasets and Benchmarks Track.

Qidan Wen, Min Hu, Maohua Lai, Juan Li, Zhenxing Hu, Kewei Quan, Jia Liu, Hua Liu, Yanbing Meng, Suling Wang, Xiaohui Wen, Chuyi Yu, Shuna Li, Shiya Huang, Yanhua Zheng, Han Lin, Xingyan Liang, Lingjing Lu, Zhefen Mai, Chunren Zhang, Taixiang Wu, Ernest H. Y. Ng, Elisabet Stener-Victorin, and Hongxia Ma. Data: Effect of acupuncture and metformin on insulin sensitivity in women with polycystic ovary syndrome and insulin resistance: A three-armed randomized controlled trial. Dryad Dataset, 2021.

Qidan Wen, Min Hu, Maohua Lai, Juan Li, Zhenxing Hu, Kewei Quan, Jia Liu, Hua Liu, Yanbing Meng, Suling Wang, Xiaohui Wen, Chuyi Yu, Shuna Li, Shiya Huang, Yanhua Zheng, Han Lin, Xingyan Liang, Lingjing Lu, Zhefen Mai, Chunren Zhang, Taixiang Wu, Ernest H. Y. Ng, Elisabet Stener-Victorin, and Hongxia Ma. Effect of acupuncture and metformin on insulin sensitivity in women with polycystic ovary syndrome and insulin resistance: A three-armed randomized controlled trial. *Human Reproduction*, 37(3):542–552, 2022. doi: 10.1093/humrep/deab272.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversations. In *Proceedings of the Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=BAakY1hNKS.

Yuzhang Xie, Xu Han, Ran Xu, Xiao Hu, Jiaying Lu, and Carl Yang. HypKG: Hypergraph-based knowledge graph contextualization for precision healthcare. In *Proceedings of the International Semantic Web Conference*, 2025. doi: 10.1007/978-3-032-09527-5_18.

Lincen Yang, Zhong Li, Matthijs van Leeuwen, and Saber Salehkaleybar. Learning subgroups with maximum treatment effects without causal heuristics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 27565–27573, 2026. doi: 10.1609/aaai.v40i32.39976. URL https://ojs.aaai.org/index.php/AAAI/article/view/39976.

Z.AI. GLM-4.5-Air model card. https://huggingface.co/zai-org/GLM-4.5-Air, 2025. Accessed 2026-05-07.

Deborah A. Zarin, Tony Tse, Rebecca J. Williams, Robert M. Califf, and Nicholas C. Ide. The ClinicalTrials.gov results database–update and key issues. *The New England Journal of Medicine*, 364(9):852–860, 2011. doi: 10.1056/NEJMsa1012065.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. AFlow: Automating agentic workflow generation. In *Proceedings of the International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=z5uVAKwmjf.

Yudi Zhang, Pei Xiao, Lu Wang, Chaoyun Zhang, Meng Fang, Yali Du, Yevgeniy Puzyrev, Randolph Yao, Si Qin, Qingwei Lin, Mykola Pechenizkiy, Dongmei Zhang, Saravan Rajmohan, and Qi Zhang. RuAG: Learned-rule-augmented generation for large language models. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=BpIbnXWfhL.

Jiehui Zhou, Linxiao Yang, Xingyu Liu, Xinyue Gu, Liang Sun, and Wei Chen. CURLS: Causal rule learning for subgroups with significant treatment effect. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4619–4630, 2024. doi: 10.1145/3637528.3671951.

OVERVIEW

In this appendix, we provide additional details and supplementary analyses for SCENE. The appendix is organized as follows. Appendix A presents extended methodological details, including scenario formalization, explicit LLM role contracts, the evolutionary grounding procedure, comprehensive experimental setups, and detailed implementation protocols for all main tables. Appendix B provides supplementary post-hoc audits (e.g., clinical effect-support geometry, L1000 inspectability, and generated feature lifecycles). Finally, Appendix C and Appendix D discuss limitations and broader impacts, respectively.

A ADDITIONAL METHOD DETAILS

A.1 NOTATION AND SCENARIO FORMALIZATION

This appendix expands the compressed methodological description in the main text. The main text defines SCENE at the level of the discovery protocol, while this section makes explicit the formal objects that are passed through the protocol. SCENE reports scenario-grounded, hypothesis-generating propositions with provenance and diagnostics; it does not report confirmed clinical recommendations, externally validated mechanisms, or claims certified by a language model.

For a biomedical scenario s, the scenario evidence is written as

$$\mathcal{D}^{(s)} = (X^{(s)}, Y^{(s)}, C^{(s)}), \tag{5}$$

where $X^{(s)}$ denotes observable features, $Y^{(s)}$ denotes outcomes or response variables used for evidence evaluation, and $C^{(s)}$ denotes task context. Biomedical prior knowledge is represented as a finite manifest of source-documented records available before discovery:

$$\mathcal{K}^{(s)} = \{\kappa_a\}_{a=1}^{A_s}, \qquad \kappa_a = (\mathrm{id}_a, u_a, \sigma_a, \chi_a, \psi_a). \tag{6}$$

Here A_s is the number of knowledge records, id_a is a stable record identifier, u_a is a compact knowledge statement, σ_a records the source descriptor, χ_a records the scenario scope, and ψ_a records the transform that made the record available to SCENE. These records provide source-traceable search cues, but they are not treated as conclusions about the current dataset.

The exported output is contextualized knowledge, represented as a set of scenario-grounded propositions:

$$\mathcal{P}^{(s)} = \{p_i\}_{i=1}^{m}, \qquad p_i = (d_i, q_i, \omega_i). \tag{7}$$

Here d_i is a bounded search direction, q_i is an executable grounded rule or finding in the current scenario, and ω_i is an evidence record containing effect, support, diagnostics, source links, split information, and provenance. The reportable unit is therefore the entire proposition p_i, not a rule or finding alone. This convention also defines the output boundary used by Figure 2: grounded rules or findings and evidence records are components of one scenario-grounded proposition rather than two independent outputs.

The main text keeps candidate search at the protocol level. Executably, the active direction and validated grounding plan induce a vocabulary $\mathcal{Z}_t$ and validity function $\Gamma_t^{(s)}$, giving

$$\mathcal{Q}_t = \{q \in \mathcal{H}(\mathcal{Z}_t) : |q| \le L, \Gamma_t^{(s)}(q) = 1, J_{\mathrm{sup}}^{(s)}(q) \ge \rho_t\}. \tag{8}$$

In the tabular instantiations used here, the default object in $\mathcal{H}(\mathcal{Z}_t)$ is a bounded conjunction

$$q(x) = \bigwedge_{j=1}^{\ell} \mathbb{I}[z_{k_j} \ \bowtie_j \ \tau_j], \qquad \ell \le L, \tag{9}$$

where literals are drawn from raw schema-visible observables or approved virtual/dynamic operators. In non-clinical settings, the same notation denotes an executable context-bounded finding rather than a clinical subgroup rule.

The scenario adapter used by Algorithm 1 can be written as

$$\mathcal{A}^{(s)} = \big(\mathcal{O}^{(s)}, \Gamma^{(s)}, J_{\mathrm{sup}}^{(s)}, J_{\mathrm{eff}}^{(s)}, \mathcal{T}^{(s)}, \Omega^{(s)}\big), \tag{10}$$

where $\mathcal{O}^{(s)}$ specifies observable families, $\Gamma^{(s)}$ validity, $J_{\mathrm{sup}}^{(s)}$ support, $J_{\mathrm{eff}}^{(s)}$ effect, $\mathcal{T}^{(s)}$ optional grounding operators, and $\Omega^{(s)}$ split, provenance, diagnostic, and output schemas.

Table 4: Notation used by SCENE.

Symbol	Meaning
s	Biomedical scenario or task instance.
$\mathcal{D}^{(s)}$	Scenario evidence, equal to $(X^{(s)}, Y^{(s)}, C^{(s)})$.
$X^{(s)}$	Observable features or measurements available under the scenario schema.
$Y^{(s)}$	Outcomes, responses, or configured evidence variables used for evaluation.
$C^{(s)}$	Task context, including scenario description, endpoint definition, and discovery constraints.
$\mathcal{K}^{(s)}$	Source-documented prior knowledge records available before discovery.
κ_a	One prior record with identifier, statement, source descriptor, scope, and transform metadata.
$\mathcal{S}^{(s)}$	Machine-readable schema with variable types, admissible windows, forbidden fields, missingness rules, and composition constraints.
d_t, η_t	Active direction and bounded search guidance at round t.
$\mathcal{Z}_t, \mathcal{G}_t$	Direction-conditioned vocabulary and grounding plan.
$q, \mathcal{Q}_t$	Candidate rule/finding and admissible candidate family.
$J_{\text{sup}}^{(s)}, J_{\text{eff}}^{(s)}$	Scenario-defined support and effect objectives.
$\mathcal{F}_t, \lambda_t$	Pareto frontier and execution log returned by the lower level.
f_t, m_t	Feedback packet and compact working memory.
$\mathcal{P}^{(s)}$	Exported set of scenario-grounded propositions.
$p_i = (d_i, q_i, \omega_i)$	One reportable proposition: direction, executable grounded object, and evidence record.
ω_i	Evidence and provenance record attached to a reported proposition.

Table 5: Scenario formalization under the common SCENE adapter interface.

Component	Clinical-trial scenario	LINCS L1000 perturbational scenario
Evidence	Subjects with admissible subgroup covariates, treatment arm, endpoint, and protocol context. Treatment and outcome fields are evaluation fields, not subgroup literals.	Perturbational profiles or signatures with cell, perturbagen, dose/time, response or connectivity score, and optional signature-derived attributes.
Candidate q	Bounded subgroup conjunction over baseline or explicitly landmarked pre-decision summaries.	Bounded context conjunction over admissible perturbation, cell, dose/time, compound, or signature attributes.
Validity	Type-correct literals, no treatment/outcome/post-outcome leakage, admissible windows, nonempty arms, and support floor.	Compatible context literals, admissible dose/time windows, matched or recurrent admissible support, and the support floor.
Support	$\min\{n_{\text{treat}}(q), n_{\text{ctrl}}(q)\}$.	Number of matched or recurrent admissible evidence instances.
Effect	Configured within-subgroup treatment-benefit contrast.	Configured connectivity or response contrast, blocked when a valid blocking design is available.
Output	Scenario-grounded proposition containing a subgroup rule, evidence record, and provenance.	Scenario-grounded proposition containing a perturbational context finding, evidence record, and provenance.

A.2 Upper-Level Controller and Role Artifacts

The upper level is implemented through typed role artifacts rather than requiring physically separate models. The same language model may instantiate multiple roles, but each role consumes and emits a different typed object. This design is intended to make the contextualization process replayable and auditable: language-model calls may propose directions, grounding maps, or critiques, but they do not directly validate biomedical claims or bypass deterministic controllers.

The accepted information flow is

$$\begin{aligned}(\mathcal{K}^{(s)}, C^{(s)}, m_{t-1}, f_{t-1}) &\xrightarrow{\text{direction planning}} (d_t, \eta_t) \xrightarrow{\text{semantic grounding}} \mathcal{G}_t \\ &\xrightarrow{\text{search}} (\mathcal{F}_t, \lambda_t) \xrightarrow{\text{feedback}} (f_t, m_t, \bar{\eta}_{t+1}).\end{aligned} \tag{11}$$

Algorithm 2 UpdateFeedback / FrontierFeedbackUpdate

Require: frontier $\mathcal{F}_t$, execution log λ_t, direction d_t, memory m_{t-1}, manifest thresholds τ
Ensure: feedback f_t, memory m_t, bounded controls $\bar{\eta}_{t+1}$

compute `invalid_rate`, `low_support_rate`, `diversity`, `diagnostic_contrast`, `alignment_gap`, and `stagnation` from $(\mathcal{F}_t, \lambda_t)$
if `leakage_flag` or `schema_violation` or `invalid_rate` $> \tau_{\text{invalid}}$ **then**
    `action` $\leftarrow$ `replace`
else if $\mathcal{F}_t = \emptyset$ **then**
    `action` $\leftarrow$ `broaden` if one manifest-permitted retry remains; otherwise `replace`
else if `low_support_rate` $> \tau_{\text{low_support}}$ or `alignment_gap` is high **then**
    `action` $\leftarrow$ `broaden`
else if `diversity` $< \tau_{\text{diversity}}$ or `stagnation` exceeds patience **then**
    `action` $\leftarrow$ `narrow`
else if `diagnostic_contrast` is high and support is adequate **then**
    `action` $\leftarrow$ `preserve` with conservative control deltas
else
    `action` $\leftarrow$ `preserve`
end if
construct f_t with the action, frontier summary, failure modes, alignment gaps, and stability notes
update m_t with retained/rejected cues, stable families, and recent failure modes
set $\bar{\eta}_{t+1}$ by clipping any control deltas to manifest bounds
return $(f_t, m_t, \bar{\eta}_{t+1})$

The upper level emits search contracts, not final biomedical conclusions. These contracts define what should be grounded next, which schema-visible objects may be used, and how the lower-level search should be bounded.

Table 6: Upper-level role artifacts and fail-closed behavior.

Artifact	Required content	Validation or fallback
Direction d_t	Search intent, supporting knowledge IDs or `model_suggested`, grounding cues, target preferences, rationale.	Unknown source IDs are rejected or marked `model_suggested`; unsupported cues cannot be reported as prior-supported.
Guidance η_t	Operator mixture, generation budget, support floor, bounds restrictions, avoid list, dynamic-feature toggles.	Numeric controls are clipped to manifest ranges; invalid toggles are ignored and logged.
Grounding plan $\mathcal{G}_t$	Matched features/windows/composites, valid literal keys, invalid combinations, seed candidates, grounding rationale.	Unmaterialized or leakage-risk literals are dropped before search.
Feedback f_t	Frontier summary, alignment gaps, failure modes, support-effect trade-offs, stability notes.	Missing diagnostics trigger conservative defaults.
Memory m_t	Retained/rejected cues, recent directions, stable families, failure modes, next-round hint.	Memory is bounded and summarized; it is not converted into external prior knowledge without a new manifest record.

The UPDATEFEEDBACK call in Algorithm 1 is implemented by the deterministic priority rule in Algorithm 2. This rule maps invalidity, support failures, diversity collapse, diagnostic contrast, alignment gaps, and stagnation into bounded next-round actions.

The compact feedback statement in the main text is therefore instantiated by a deterministic priority order. Default thresholds are recorded in the manifest, e.g., diagnostic-contrast high/low cutoffs, diversity floor, invalid-rate ceiling, low-support ceiling, and upper-level stagnation patience. These values may be tuned per scenario before discovery, but not by final held-out or test metrics.

Language-model calls are validated as JSON-producing proposal functions. A strategist artifact contains `direction`, `knowledge_ids`, `grounding_cues`, `target_preference`, `control_adjustment`, and `rationale`. A proposer artifact contains `semantic_matches`, `grounding_map`, `candidate_literals`,

`invalid_combinations`, `seed_candidates`, and optional `feature_proposals`. Critic/refiner artifacts contain frontier summaries, failure modes, stability notes, next-direction hints, memory updates, and bounded control deltas. Extra top-level fields, unknown feature identifiers, unregistered source IDs, non-finite numbers, and claims of validation success are rejected.

A.3 Role Instruction Contracts

SCENE specifies role instructions as contracts rather than as unconstrained prompts. Each language-model call is treated as a JSON-producing proposal function: the model may suggest a direction, grounding, critique, or refinement, but its output has no executable effect until parsed, checked against the frozen manifest, and validated by deterministic controllers. The controller supplies only discovery-visible context, including source records, schema summaries, frontier feedback, split-visibility flags, and bounded control ranges. It rejects malformed JSON, unknown source identifiers, unknown feature keys, forbidden fields, non-finite values, out-of-range controls, and unsupported biomedical assertions. The following compact contracts summarize the role interfaces used by SCENE; scenario-specific prompts instantiate the placeholders with manifest-visible context, but the executable boundary is the schema and controller rather than the wording of the prompt.

Strategist contract

```
role: strategist
input:
{
  "scenario_context": "<task and discovery-visible split context>",
  "knowledge_records": [{"id": "kappa_a", "cue": "..."}],
  "schema_summary": {"feature_families": ["..."], "forbidden": ["..."]},
  "memory": "<bounded prior-round summary>",
  "frontier_summary": "<recent frontier status and trade-offs>",
  "control_bounds": "<manifest ranges>"
}
instructions:
- Propose a search direction, not a biomedical conclusion.
- Use supplied knowledge IDs only; otherwise mark "model_suggested".
- Map concepts to schema-visible grounding cues or mark them unresolved.
- Propose a replacement or revised direction when frontier feedback
    indicates failure.
- Do not report effect size, statistical significance, or validation
    status.
output_json:
{
  "direction": "...",
  "status": "source_supported|model_suggested|feedback_derived",
  "knowledge_ids": ["kappa_a"],
  "grounding_cues": ["..."],
  "target_preference": {"prefer": ["..."], "avoid": ["..."]},
  "control_adjustment": [{"param": "...", "delta": 0.0, "reason": "..."}],

  "rationale": "short evidence-bounded rationale"
}
```

Proposer-grounder contract

```
role: proposer_grounder
input:
{
  "direction": "<selected direction>",
  "schema_summary": "<feature keys, types, windows, grammar>",
  "allowed_operators": ["mutation", "crossover", "tuning", "injection"],
  "forbidden_fields": ["..."],
  "feature_statistics": "<discovery-visible summaries only>"
}
instructions:
- Use only supplied feature keys, windows, and operators.
```

```
- Do not use treatment, outcome, post-outcome, or
  validation-derived fields as literals.
- Translate each cue into matched features, admissible windows,
  candidate literals, or an explicit invalid/unresolved entry.
- Do not claim effect, significance, or validation success.
output_json:
{
  "semantic_matches": [{"cue": "...", "features": ["..."], "windows":
    ["..."]}],
  "grounding_map": {"cue_id": {"literal_keys": ["..."], "status": "
    matched"}},
  "candidate_literals": [{"feature": "...", "op": ">|<=", "value_source":
     "grid"}],
  "seed_candidates": [["literal_key_a", "literal_key_b"]],
  "invalid_combinations": [{"cue": "...", "reason": "not_schema_visible
    "}],
  "feature_proposals": [{"request": "...", "reason": "..."}],
  "safety": {"uses_forbidden_field": false}
}
```

Critic-refiner contract

```
role: critic_refiner
input:
{
  "direction": "<current direction>",
  "frontier_summary": "<Pareto set, support/effect, diversity>",
  "execution_log": "<invalid, low-support, weak-alignment cases>",
  "current_controls": "<bounded runtime controls>"
}
instructions:
- Use only frontier and execution-log evidence; do not invent experience.
- Summarize failure_modes and stability_notes separately.
- Choose one next action: preserve, narrow, broaden, or replace.
- Keep control_delta bounded, conservative, and next-round only.
- Do not convert feedback into external prior knowledge.
output_json:
{
  "frontier_summary": {"status": "coherent|empty|unstable|low_support"},
  "failure_modes": ["..."],
  "stability_notes": ["..."],
  "action": "preserve|narrow|broaden|replace",
  "next_direction_hint": "...",
  "control_delta": [{"param": "...", "delta": 0.0, "reason": "..."}],
  "memory_update": "bounded summary"
}
```

When `feature_proposals` are present, the scenario adapter may request one virtual or dynamic feature plan using only registered tables, allowed variables, admissible windows, and leakage-safe operators; invalid or unmaterialized plans are dropped and logged. If executable synthesis is used for an accepted feature plan, it is separately sandboxed and validated before generated features enter search.

A.4 Role Artifact Templates and One-Round Trace

The following schematic artifact illustrates the role interface using a simplified metformin-response example. Field values are illustrative rather than experimental observations, and the object is intended to show the replay contract rather than measured support or effect. The example emphasizes that a reported output is a proposition-level object containing a direction, an executable grounded object, and an evidence record.

Metformin-response example

```
{
  "knowledge_record": {
    "id": "kappa_metabolic_01",
    "cue": "metabolic dysregulation may modify metformin response",
    "scope": "clinical subgroup discovery"
  },
  "direction": {
    "text": "search for an insulin-resistance or metabolic-risk subgroup",

    "knowledge_ids": ["kappa_metabolic_01"],
    "status": "source_supported"
  },
  "semantic_match": {
    "features": ["baseline_BMI", "fasting_glucose", "HOMA_IR_if_available
    "],
    "windows": ["baseline"],
    "dropped": ["post-treatment glucose"]
  },
  "search_guidance": {
    "support_floor": "manifest_min_balanced_support",
    "operators": ["mutation", "crossover", "tuning"],
    "avoid": ["treatment", "outcome", "post-outcome"]
  },
  "grounded_object": {
    "candidate_key": "baseline_BMI>tau_1 AND fasting_glucose>tau_2",
    "threshold_source": "predeclared discovery grid"
  },
  "frontier_log": {
    "frontier_status": "coherent",
    "failure_modes": ["HOMA_IR unavailable in this schema"]
  },
  "feedback": {
    "action": "preserve",
    "memory_update": "metabolic baseline features remain productive"
  },
  "reported_proposition": {
    "direction": "search for an insulin-resistance or metabolic-risk
    subgroup",
    "grounded_object": "baseline_BMI>tau_1 AND fasting_glucose>tau_2",
    "evidence_record": {
      "support": "...",
      "effect": "...",
      "source_id": "kappa_metabolic_01",
      "NoLeak": true,
      "split_id": "...",
      "diagnostics": "..."
    }
  }
}
```


A one-round trace is therefore an artifact chain rather than a transcript of agent conversation:

$$\kappa_a \to d_t \to (\mathcal{G}_t, \eta_t) \to (\mathcal{F}_t, \lambda_t) \to f_t \to p_i. \tag{12}$$

The knowledge record supplies a provenance-bearing cue; the direction states a bounded search intent; schema match and guidance translate that intent into executable schema objects; the frontier and log expose what the lower level could or could not ground; feedback updates memory and controls; and the final arrow emits a proposition-level object $p_i = (d_i, q_i, \omega_i)$ only if the candidate also satisfies the reporting predicate in Appendix A.7. This trace is illustrative of the interface contract and does not encode a measured effect size, support count, or empirical success claim.

Algorithm 3 Direction-conditioned grounding at round t

Require: direction d_t, contract $\mathcal{G}_t = (\mathcal{Z}_t, \Gamma_t^{(s)}, \mathcal{C}_{t,0})$, evidence $\mathcal{D}^{(s)}$, adapter $\mathcal{A}^{(s)}$, controls $\eta_t = (\pi_t, H_t, \rho_t, \beta_t)$
Ensure: frontier $\mathcal{F}_t$ and execution log λ_t
freeze replay configuration ξ_t ▷ seed, split, adapter, grammar, controls
for $g \leftarrow 0$ to H_t **do**
  $\hat{\mathcal{C}}_{t,g} \leftarrow \textsc{Validate}(\mathcal{C}_{t,g}, \Gamma_t^{(s)}, \rho_t)$ ▷ deduplicate and filter
  $\mathcal{R}_{t,g} \leftarrow \textsc{Evaluate}(\hat{\mathcal{C}}_{t,g}, \mathcal{D}^{(s)}, \mathcal{A}^{(s)})$ ▷ effect/support diagnostics
  update archive $\mathcal{R}_t$ and log λ_t with kept, rejected, and invalid cases
  if $g < H_t$ **then**
    $\mathcal{U}_{t,g} \leftarrow \textsc{Vary}(\hat{\mathcal{C}}_{t,g}, d_t, \mathcal{G}_t, \pi_t)$ ▷ mutation, crossover, tuning, injection
    $\mathcal{C}_{t,g+1} \leftarrow \textsc{Select}(\hat{\mathcal{C}}_{t,g} \cup \mathcal{U}_{t,g}, \mathcal{R}_t, \eta_t)$ ▷ rank, diversity, key order
  end if
end for
return $\mathcal{F}_t \leftarrow \mathrm{Pareto}(\mathcal{R}_t)$, λ_t

A.5 Evolutionary Grounding Procedure

The grounding loop executes the proposer's search space under the current direction. It maintains a population $\mathcal{C}_{t,g}$, an archive $\mathcal{R}_t$, and an append-only log λ_t. The elements of this population are executable candidates: clinical subgroup rules in clinical tasks and context-bounded findings in L1000 tasks. Pareto dominance uses only $(J_{\mathrm{eff}}^{(s)}, J_{\mathrm{sup}}^{(s)})$; pseudo-validation can diagnose instability or break exact ties after Pareto rank and diversity, but it cannot create dominance.

Table 7: Executable grounding controls recorded in ξ_t and λ_t.

Component	Operation	Logged state
Initialization	Mix proposer seeds with admissible diversity-oriented candidates.	Seed source, validity status, duplicates, support failures.
Mutation	Add, drop, or modify literals, thresholds, bins, dose/time windows, or landmark windows.	Parent key, edited field, grid identifier, validity reason.
Crossover	Combine compatible partial structures from frontier or archive candidates.	Parent keys, compatibility checks, resulting semantic key.
Tuning	Move thresholds/windows along a predeclared grid around supported candidates.	Old/new values, grid step, support/effect change.
Injection	Introduce fresh candidates from grounding cues, feature proposals, or refiner hints.	Trigger, source cue, source record or `model_suggested` tag.
Selection	Apply validity, deduplication, support floor, Pareto rank, diversity, deterministic key order.	Full ordering key and retained/rejected status.
Replay	Freeze seed stream, split ID, adapter version, candidate grammar, population size, archive cap.	Configuration hash and role-output hash when available.

Scenario manifests may set stricter floors or disable operators, but final validation metrics cannot alter search defaults for the same run.

A.6 Grounding Operators

Some directions refer to concepts that are only partially visible in raw observables. This is the main operational challenge of knowledge contextualization: broad biomedical ideas are rarely expressed in the same vocabulary as the dataset schema. SCENE can therefore expand the grounding vocabulary from $\mathcal{Z}^{(0)}$ to

$$\mathcal{Z}_t = \mathcal{Z}^{(0)} \cup \mathcal{Z}^{(\mathrm{virt})} \cup \mathcal{Z}^{(\mathrm{dyn})}. \tag{13}$$

Here $\mathcal{Z}^{(0)}$ contains raw schema-visible observables, $\mathcal{Z}^{(\mathrm{virt})}$ contains approved virtual composites, and $\mathcal{Z}^{(\mathrm{dyn})}$ contains approved dynamic or windowed summaries. The executable grammar is fixed

Table 8: Default executable values used when a scenario manifest does not specify stricter settings.

Field	Default	Role in execution
Population N_t	Clinical: 240; L1000: 160; smoke: 40	Candidate population and offspring budget per generation after clipping to $[40, 320]$.
Generations H_t	Clinical: 20; L1000: 14; smoke: 2	Full runs are clipped to $[5, 40]$; smoke mode is flagged as non-evaluation.
Rule length and support	$L = 3$; clinical $\rho_t \geq 30$ balanced-arm support; L1000 $\rho_t \geq 60$ instances	Scenario adapters may raise floors; broadening cannot lower them below the manifest floor.
Operator mix π_0	Mutation 0.60, crossover 0.25, injection 0.15	Each coordinate is clipped to $[0.05, 0.80]$ and renormalized.
Retry/archive	Retry cap $5N_t$; archive cap $\min(1000, 5N_t)$	Invalid, duplicate, or low-support offspring are retried; archives are pruned by Pareto rank, diversity, then key order.
Stagnation	5 generations lower-level; 3 rounds upper-level	Triggers early stopping below or direction replacement above.
Controller thresholds	$\Delta_{\text{high}} = 0.65$, $\Delta_{\text{low}} = 0.25$, diversity floor 0.20, invalid-rate ceiling 0.50, low-support-rate ceiling 0.60	Used for preserve/narrow/broaden/replace hints before final validation is visible.

by the scenario manifest before discovery:

$$\begin{aligned} q &::= \ell_1 \wedge \cdots \wedge \ell_r, \qquad 1 \leq r \leq L, \\ \ell &::= z \bowtie \tau \mid z \in B \mid \text{missing}(z) = b, \\ z &::= x_k \mid \text{virt}_r(x_{a_1}, \ldots, x_{a_m}) \mid \text{dyn}_r(x_{a_1}, \ldots, x_{a_m}; w), \\ \bowtie &\in \{<, \leq, >, \geq\}. \end{aligned} \tag{14}$$

Here x_k must be a schema-visible observable, B and τ must come from predeclared grids or manifest-approved category sets, w is an admissible temporal or perturbational window, and every operator instance must return a typed feature with provenance. Mutation, crossover, tuning, and injection can edit only objects generated by this grammar; invalid literals are dropped before scoring and recorded in λ_t.

Table 9: Examples of how broad biomedical cues can be grounded under different scenario schemas. These examples illustrate the grounding interface; they are not additional rules or reported findings.

Broad cue	Clinical-trial grounding	L1000 grounding	Gate
Metabolic dysregulation	BMI, fasting glucose, triglycerides, HOMA-IR, insulin-related summaries when available.	Pathway or signature summaries related to metabolic programs when materialized from registered profiles.	Schema-visible variables only; no outcome or validation-derived fields.
Treatment-response heterogeneity	Baseline or landmarked pre-decision subgroup literals, with treatment and endpoint reserved for evaluation.	Context literals over cell, perturbagen, dose, time, and admissible signature attributes.	Clinical treatment/outcome fields cannot become literals.
Dynamic response context	Predeclared longitudinal or trajectory summaries within admissible windows.	Dose-time summaries or perturbational signature summaries within allowed windows.	Window must be declared before search and replayable on holdout.
Mechanistic assay context	Available only when represented by baseline, laboratory, pathology, or approved derived measurements.	Cell-line context, perturbagen class, pathway summaries, and target-compatible signature attributes.	Comparison pools and target-derived labels cannot be constructed post hoc.

Virtual operators derive scalar or categorical features from existing tables, such as thresholded composites, ratios, bins, or feature-family summaries. Dynamic operators summarize auxiliary

longitudinal or perturbational traces, such as pre/post changes, event counts, window aggregates, dose-time summaries, or perturbational signature summaries. Both return typed features with construction provenance, missingness indicators, admissibility tags, and fail-closed behavior. A proposed feature that cannot be materialized from available columns, violates the schema, or uses forbidden response/validation information is excluded and logged. Thus, generated features expand the grounding space but do not relax the executable boundary.

A.7 Reporting, Provenance, and Split Boundaries

A candidate may guide feedback even when it is not reportable, provided it was produced from discovery-visible information. Export is stricter than feedback: a candidate becomes a reported scenario-grounded proposition only if it has frontier membership, validity, support, provenance, leakage checks, and diagnostics. We write the reporting predicate as

$$\begin{aligned}\mathrm{Report}(q, d_t, t) = \mathbf{1}\{&q \in \mathcal{F}_t,\ \Gamma_t^{(s)}(q) = 1,\\ &J_{\mathrm{sup}}^{(s)}(q) \geq \rho_t,\ \mathrm{Prov}(q, d_t, t),\\ &\mathrm{NoLeak}^{(s)}(q),\ \mathrm{Diag}(q, t)\}.\end{aligned} \tag{15}$$

When this predicate is satisfied, SCENE exports a proposition

$$p_i = (d_t, q, \omega_i), \tag{16}$$

where ω_i stores the evidence and provenance fields. If the predicate fails, the candidate may remain in the execution log for feedback, but it is not exported as contextualized knowledge.

Table 10: Field-level checks required before a candidate can be exported as a scenario-grounded proposition.

Check	Required fields	Fail-closed behavior
Provenance	Candidate, round, direction, manifest, schema, split, replay-config IDs; knowledge-record IDs or `model_suggested`; grounding map; operator provenance.	Missing IDs, hashes, grounding entries, or operator provenance prevent export.
No leakage	Forbidden-field list, literal keys, operator input columns, split lineage, L1000 comparison-pool lineage.	Clinical treatment/outcome/post-outcome literals fail. L1000 response/connectivity-derived, validation-derived, comparison-pool, or post hoc block features fail.
Diagnostics	Effect/support, frontier membership, Pareto rank or archive status, diagnostic contrast, pseudo-validation status or missingness, blocking status where relevant, invalid/low-support counts.	Missing diagnostics prevent export, although the case may remain in λ_t for feedback.

Split lineage is frozen before discovery. Source-documented priors may be consumed by strategist and proposer; within-run memory and logs may guide the same discovery episode; discovery-visible pseudo-validation may guide conservative bounded control update; final held-out or test archives are post-discovery audit artifacts only. Importing a completed validation summary into a later run requires a new frozen knowledge record with source run, split ID, timestamp, evidence-status label, and allowed consumer roles, so that validation-derived knowledge is explicit rather than silently recycled.

A run manifest records scenario ID, freeze time, schema hash, split ID, controller thresholds, control bounds, allowed feature families, forbidden fields, enabled operators, role-output hashes when available, adapter version, and final-validation boundary. Source-free suggestions may be logged as `model_suggested` only for data-grounded search hypotheses; they cannot be cited as external biomedical knowledge or prior-supported mechanisms.

Minimal manifest fields. An executable run stores a compact JSON-like manifest before discovery:

Manifest contract

```
{
  "scenario_id": "clinical_trial_or_l1000_task",
  "freeze_time_utc": "...",
  "schema_hash": "sha256:...",
  "split_id": "discovery_split_seed",
  "knowledge_records": [{"record_id": "...", "source_type": "..."}],
  "controller_policy": {
    "action_priority": ["replace", "broaden", "narrow", "preserve"],
    "delta_high": 0.65,
    "diversity_floor": 0.20,
    "invalid_rate_ceiling": 0.50
  },
  "control_bounds": {
    "population_size": [40, 320],
    "generations": [5, 40],
    "operator_probability": [0.05, 0.80],
    "max_rule_len": 3
  },
  "final_validation_boundary": {
    "visible_to_discovery": false,
    "metrics_feed_back": false
  }
}
```

Actual manifests additionally store scenario-specific paths, hashes, replay seeds, adapter versions, enabled operators, forbidden fields, and role-output hashes when role outputs are cached.

A.8 Scenario-Specific Estimators

The scenario adapter defines the concrete effect and support functions used by lower-level search. The main text presents these functions abstractly as $J_{\text{eff}}^{(s)}$ and $J_{\text{sup}}^{(s)}$; this section records the two main instantiations used in the paper.

For clinical treatment-benefit discovery and a fixed admissible rule q, the configured effect is a descriptive bad-outcome reduction:

$$\begin{aligned} J_{\text{eff}}^{(\text{trial})}(q) = {} & \Pr(Y_{\text{bad}} = 1 \mid q(X) = 1, T = 0) \\ & - \Pr(Y_{\text{bad}} = 1 \mid q(X) = 1, T = 1), \\ J_{\text{sup}}^{(\text{trial})}(q) = {} & \min\{n_1(q), n_0(q)\}. \end{aligned} \tag{17}$$

This sign convention matches the main clinical endpoint: larger values indicate a larger reduction in unfavorable outcomes under the treatment arm relative to the control arm. Because SCENE adaptively searches over many candidates and refines later directions using discovery feedback, reported subgroup effects are exploratory, selection-conditioned estimates. Unless a separate pre-registered or multiplicity-adjusted analysis is performed, SCENE does not report valid post-selection p-values, treatment-labeling claims, or external clinical recommendations. Rules using post-baseline summaries are marked as predictive or associational unless a valid landmark interpretation is declared by the scenario.

For LINCS L1000, the configured effect is a context-versus-background target-response or connectivity contrast. The adapter freezes blocking variables, comparison-pool eligibility, exclusion lists, threshold grids, and feature-construction rules before search. A candidate may determine selected-instance membership through its admissible literals, but it may not determine controls after observing its effect. Let c_i be the response or connectivity score. For blocking strata $\mathcal{B}(q)$ with both selected and comparison evidence,

$$\begin{aligned} \mathcal{M}_b(q) &= \{i \in \mathcal{M}(q) : b(i) = b\}, \\ \mathcal{M}_b^0(q) &= \{i \in \mathcal{M}^0(q) : b(i) = b\}, \end{aligned} \tag{18}$$

with weights $w_b(q) = |\mathcal{M}_b(q)| / \sum_{b' \in \mathcal{B}(q)} |\mathcal{M}_{b'}(q)|$. The blocked contrast is

$$J_{\text{eff}}^{(\text{l1000,blk})}(q) = \sum_{b \in \mathcal{B}(q)} w_b(q) \left(\bar{c}_{\mathcal{M}_b(q)} - \bar{c}_{\mathcal{M}_b^0(q)} \right). \tag{19}$$

If no admissible blocking design is available, the adapter uses the predeclared unblocked fallback $\bar{c}_{\mathcal{M}(q)} - \bar{c}_{\mathcal{M}^0(q)}$ when support conditions are met and records `blocking_status=unblocked`. L1000 findings are descriptive perturbational associations within the configured assay and matching design, not causal compound effects or validated mechanisms.

A.9 Experiment Matrix and Reporting Contract

This subsection links the method objects defined above to the three empirical reporting settings. It does not introduce additional search rules; detailed task construction and metric definitions are deferred to Appendices A.10–A.13. SCENE is evaluated in three settings: (i) clinical subgroup discovery, (ii) perturbational mechanism discovery, and (iii) downstream few-shot contextual knowledge reuse. These settings share the same high-level discovery backbone but differ in unit of analysis, target variable, candidate language, and evaluation functional.

Clinical subgroup discovery. The unit is one patient. Each run operates on one clinical task frame, one paired train–holdout split, and one discovery seed. Each method returns an ordered rule list, but only the first committed rule is used in the paper-facing best-1 benchmark. Holdout outcomes are never used to choose or reorder rules. Under the proposition notation above, the grounded object q_i is a subgroup rule and ω_i stores the held-out replay evidence, support, diagnostics, and provenance.

Perturbational mechanism discovery. The unit is one L1000 signature instance. Each run fixes one target mechanism, one split, one search seed, and one ablation profile. Discovery is executed on the training split only; all reported perturbational metrics are computed by replaying frozen rules on the held-out split. Here q_i is a context-bounded perturbational finding, and ω_i records target-response evidence, support, replay diagnostics, and split lineage.

Contextual knowledge reuse. The unit is one downstream tabular row, either a patient or a signature instance. Each run fixes one scene, one outer split, one few-shot exemplar set, one prompt condition, and one deterministic LLM decoding policy. Discovered propositions are converted into compact knowledge cards only from train-side discovery and selected only on validation data. Test rows are used only once for final prediction and metric computation. This setting evaluates whether proposition-level outputs can be reused as auxiliary context; it is not an independent validation of every biomedical mechanism encoded in a card.

Across all three settings, we maintain strict train/validation/test separation: held-out outcomes, held-out connectivity, and final downstream labels are not used to revise prompts, re-select rules, or change knowledge-card content within the same evaluation episode.

A.10 Clinical Task Definitions

Table 11 expands the short clinical task headers used in Table 1. The six tasks are constructed from two de-identified patient-level trial tables associated with the registered trials in the table; ClinicalTrials.gov provides identifiers and protocol context, not patient-level modeling rows (ClinicalTrials.gov, 2011; 2022; Zarin et al., 2011). The trials were selected by pre-analysis feasibility and coverage criteria: randomized arms, patient-level covariates, interpretable treatment/endpoints, enough support for repeated replay, and coverage of more than one biomedical setting. No trial was added or removed after inspecting SCENE held-out performance. The NCT00174655 oncology table is accessed through Project Data Sphere (Project Data Sphere, 2026; Green et al., 2015); the NCT02491333 PCOS table is derived from the Dryad release of Wen et al. (Wen et al., 2021; 2022). These are treatment-benefit subgroup-discovery tasks: each asks whether a prioritized subgroup rule identifies patients for whom treatment reduces a task-specific unfavorable clinical outcome, rather than estimating a global average treatment effect.

Table 11: Clinical task definitions used in the treatment-benefit subgroup-discovery benchmark. BL/TR denote baseline/static and trajectory-enhanced feature views.

Header	Trial and clinical setting	Treatment question and feature view	Response or unfavorable-outcome definition
B-D-BL	NCT00174655; node-positive operable breast cancer.	Doxorubicin-containing chemotherapy-schedule comparison using baseline/static covariates.	Bad outcome is recurrence or death by the preprocessed disease-free-survival (DFS) status.
B-D-TR	NCT00174655; node-positive operable breast cancer.	Same doxorubicin-schedule comparison using trajectory-enhanced features.	Bad outcome is recurrence or death by the preprocessed DFS status.
P-M-BL	NCT02491333; polycystic ovary syndrome (PCOS) under sham acupuncture.	Metformin versus placebo using baseline/static covariates.	Good response is month-4 HOMA-IR reduction of at least 25% from baseline; bad outcome is failing this threshold.
P-M-TR	NCT02491333; PCOS under sham acupuncture.	Metformin versus placebo using trajectory-enhanced features.	Good response is month-4 HOMA-IR reduction of at least 25% from baseline; bad outcome is failing this threshold.
P-A-FPG	NCT02491333; PCOS under placebo metformin.	Acupuncture-response subgroup discovery for fasting-plasma-glucose improvement.	Good response is month-4 fasting plasma glucose reduction of at least 0.2 mmol/L from baseline; bad outcome is failing this threshold.
P-A-AUC	NCT02491333; PCOS under placebo metformin.	Acupuncture-response subgroup discovery for OGTT glucose-AUC improvement.	Good response is month-4 OGTT glucose-AUC reduction of at least 1.0 from baseline; bad outcome is failing this threshold.

These definitions determine the task-specific held-out endpoint used by the main clinical endpoint, $\mathrm{ARR}_{\mathrm{bad}}$. Treatment, outcome, post-outcome, identifier, and bookkeeping fields are excluded from candidate subgroup features before fitting any method.

A.11 Implementation Details for Table 1

Table 1 is implemented as a *best-1 rule-selection benchmark*, following the convention that exploratory subgroup findings should be fixed before independent assessment (Rothwell, 2005; Kent et al., 2020). The unit of evaluation is not an unranked pool of discovered subgroups, but the first rule that a method commits to before held-out outcomes are inspected. For clinical task frame d, split identifier b, and method m, the discovery code exports an ordered rule list

$$\mathcal{R}_{m,d,b} = \left(q^{(1)}_{m,d,b}, q^{(2)}_{m,d,b}, \dots\right). \tag{20}$$

The reported rule for that method–task–split is

$$q^{\star}_{m,d,b} = q^{(1)}_{m,d,b}. \tag{21}$$

The held-out partition is used only after $q^{\star}_{m,d,b}$ has been fixed. We do not choose the rule with the best test-set risk difference, and we do not average multiple rules from a single method–split run. This implementation matches the scientific use case of SCENE: the system should return prioritized, interpretable subgroup hypotheses whose held-out replay can be audited, rather than producing a large catalogue from which favorable test-set rules are selected post hoc.

Task frames and endpoint construction. Table 1 uses the six paper-facing clinical reporting frames `B-D-BL`, `B-D-TR`, `P-M-BL`, `P-M-TR`, `P-A-FPG`, and `P-A-AUC`. These headers follow Table 11: `B-D-BL`/`B-D-TR` are baseline/static and trajectory-enhanced breast-cancer doxorubicin-schedule

frames; `P-M-BL`/`P-M-TR` are baseline/static and trajectory-enhanced PCOS metformin-response frames; `P-A-FPG` and `P-A-AUC` are PCOS acupuncture-response frames for fasting-plasma-glucose and OGTT-AUC improvement. Each modeling table contains one row per patient, so row-level splitting is patient-level splitting. The treatment indicator is the task-specific randomized arm encoded in the corresponding modeling frame. The favorable-response indicator is constructed as follows. Let $\mathcal{D}_{\mathrm{DFS}} = \{\text{B-D-BL}, \text{B-D-TR}\}$ and $\mathcal{D}_{\mathrm{HOMA}} = \{\text{P-M-BL}, \text{P-M-TR}\}$. Let F_i^{DFS} denote the preprocessed disease-free-survival indicator, with $F_i^{\mathrm{DFS}} = 1$ indicating no recurrence or death. We write r_i^{HOMA} for the month-4 relative HOMA-IR reduction, and a_i^{FPG} and a_i^{AUC} for the corresponding absolute reductions in fasting plasma glucose and OGTT glucose AUC. The task-specific favorable endpoint is

$$G_{i,d} = \begin{cases} F_i^{\mathrm{DFS}}, & d \in \mathcal{D}_{\mathrm{DFS}}, \\ \mathbf{1}\{r_i^{\mathrm{HOMA}} \geq 0.25\}, & d \in \mathcal{D}_{\mathrm{HOMA}}, \\ \mathbf{1}\{a_i^{\mathrm{FPG}} \geq 0.2 \text{ mmol/L}\}, & d = \text{P-A-FPG}, \\ \mathbf{1}\{a_i^{\mathrm{AUC}} \geq 1.0\}, & d = \text{P-A-AUC}. \end{cases} \tag{22}$$

The bad-outcome endpoint used by the Table 1 risk-difference estimator is

$$Y_{i,d}^{\mathrm{bad}} = 1 - G_{i,d}. \tag{23}$$

Treatment, outcome, death, post-outcome, identifier, and bookkeeping columns are excluded from the feature set before fitting any method. Numeric imputers, categorical encoders, binarization maps, and threshold grids are fit on the training partition and replayed on the held-out partition. The response thresholds are fixed before method comparison and are used only to define task endpoints; they are not tuned using SCENE or baseline held-out performance.

Split manifest and pairing. The reporting run repeats the complete rule-discovery and best-1 replay pipeline over $B = 50$ patient-level split identifiers. Each split uses a 75/25 train/test partition, i.e. `test_size=0.25`, stratified by treatment arm and bad-outcome status when feasible. The benchmark runner supports either an explicit comma-separated seed list or automatic seed generation

$$s_b = s_0 + (b-1)\Delta_s, \qquad b = 1, \ldots, B. \tag{24}$$

For the Table 1 reporting run, we use the manifest seed list induced by `base_seed=58` and the runner seed policy recorded in the output manifest. When the default benchmark policy is used, this gives $s_b = 58 + 10(b-1)$; if an explicit seed list is supplied, that list overrides `-num-runs`. The renderer materializes a split identifier as `split_id=seed_<seed>`: for baselines the seed is parsed from `run_<id>_seed_<seed>`, and for SCENE it is read from the archived `config_snapshot.json`. The final paper rendering requires the paired equality

$$\mathcal{I}_{m,d} = \mathcal{I}_d \quad \text{for all methods } m \text{ on task frame } d, \tag{25}$$

where $\mathcal{I}_{m,d}$ is the set of used split identifiers for method m on task d. A rendered table is considered reportable only when the input manifest contains the expected $B_{m,d} = 50$ used best-1 artifacts for every method–task row and the split sets satisfy this equality. Partial diagnostic renderings with smaller, uneven, or unpaired $B_{m,d}$ are useful for debugging, but the final paper rendering is configured to abort rather than silently skip or impute rows.

Table 12: Artifact contract used by the Table 1 renderer. The contract is designed to make the best-1 boundary and the paired-split accounting auditable from files, not from prose alone.

Source	Best-1 rule source	Required audit fields
Baselines	First row of the exported rule CSV after training-side ranking.	Method, task, run index, seed, source path, required test columns, used/skipped status.
SCENE	First validation row after SCENE discovery-side Pareto/export ordering.	Task header, feature-view/dynamic flag, run index, seed, source path, required test columns, used/skipped status.
Renderer outputs	The run-level file is the single source for summary aggregation.	$B_{m,d}$, split identifier coverage, metric columns, skipped-file reasons.

SCENE validation artifacts. SCENE entries are read from post-discovery replay files rather than from training-side search logs. For `B-D-BL` and `B-D-TR`, the renderer uses the baseline/static and trajectory-enhanced validation outputs under the NCT00174655 SCENE root. For `P-M-BL`, `P-M-TR`, `P-A-FPG`, and `P-A-AUC`, it uses the corresponding NCT02491333 validation roots. The SCENE columns consumed by the renderer are `test_subgroup_size_pct`, `test_bad_rate_treat`, `test_bad_rate_ctrl`, `test_rd_bad`, `test_rd_bad_ci_low`, and `test_rd_bad_ci_high`. The SCENE clinical implementation uses population size 240, 20 generations, maximum rule length 3, minimum discovery-side subgroup size 30, evolutionary-loop V2, and loop-stage virtual feature construction, with the dynamic-feature switch enabled for `B-D-TR`, `P-M-TR`, `P-A-FPG`, and `P-A-AUC`, and disabled for `B-D-BL` and `P-M-BL`. These settings are fixed before held-out replay. The validation files are held-out replay artifacts: they are not fed back into the same discovery episode and are not used to revise the rule, direction, feature proposals, or operator mixture for that split.

Agent and compute resources. The six data-only Table 1 baselines are CPU-side tabular learners or rule-mining procedures. SCENE's agent-facing steps and RuAG's feature selection use third-party LLM API routes rather than local GPU inference; the logged configuration records the provider, model identifier, decoding settings, retry policy, and returned role artifacts. Rule replay, risk-difference estimation, paired-split checks, and final table rendering are performed from cached artifacts on CPU workers. CPU-side aggregation was run on a single CPU workstation (AMD Ryzen 7 7840H, 16 GB RAM), no local GPU is required once the archived SCENE artifacts are available. The Table 1 CPU-side replay and rendering jobs complete within a few CPU-hours after the processed modeling tables are available, excluding provider-dependent LLM API latency.

Baseline wrappers and fixed knobs. The six data-only baselines cover classical subgroup discovery and recent rule-learning baselines: Virtual Twins (Foster et al., 2011), SIDES (Lipkovich et al., 2011), honest causal trees (Athey & Imbens, 2016), causal forests (Athey et al., 2019), CURLS (Zhou et al., 2024), and MaxTE Subgroups (Yang et al., 2026). All wrappers receive the same task list, endpoint definition, leakage exclusions, split identifier, test fraction, minimum arm-support policy when available, and rule-complexity budget. Each wrapper converts the underlying method into the same exported object: a ranked list of readable subgroup rules and held-out replay columns for those rules.

MaxTE + Grounding uses the same MaxTE wrapper with the feature restriction specified in Appendix B.9; the command below describes the original six data-only baselines.

The runner writes baseline files under `runs/<method>/run_<index>_seed_<seed>/`. The command below is the reporting form of the baseline execution; a shorter command with fewer runs is used only for sanity checks.

Baseline benchmark command

```
& ".\.venv_curls_repro\Scripts\python.exe" ".\analysis\
    quantitative_benchmark_runner.py" `
  --methods "virtual_twins,sides,honest_causal_tree,causal_forest,curls,
    MaxTE_subgroups" `
  --datasets "B-D-BL,B-D-TR,P-M-BL,P-M-TR,P-A-FPG,P-A-AUC" `
  --num-runs 50 `
  --base-seed 58 `
  --seed-step 10 `
  --test-size 0.25 `
  --rule-feature-cap 5 `
  --analysis-dir ".\analysis" `
  --out-dir $Out `
  --rscript-path "$RscriptExe" `
  --curls-root ".\CURLS\osfstorage-archive"
```

If a method produces no valid rule or a matched file lacks required columns, the file is recorded in `table1_input_manifest.csv` or `table1_skipped_files.csv` with a reason and does not silently contribute imputed metrics. With `-expected-runs 50`

Table 13: Method-specific implementation details for the six data-only Table 1 baselines. The values shown are fixed before held-out replay and are not tuned using Table 1 outcomes.

Method	Wrapper implementation	Main fixed knobs	Rule ordering before replay
Virtual Twins	Separate treated/control random-forest classifiers for bad outcome, followed by a regression-tree surrogate over predicted benefit.	RF estimators $= 300$, RF leaf size $= 5$, surrogate max depth from rule cap, surrogate min leaf $= 30$, minimum arm support $= 8$.	Predicted benefit, training support, and lower training-side $\mathrm{RD}_{\mathrm{bad}}$.
SIDES	Local SIDES-style beam search over encoded categorical literals and numeric thresholds.	Beam width $= 12$, max feature pool $= 60$, max thresholds per feature $= 5$, min node size $\max(30, 2\times$ min arm support$)$.	Training-side SIDES score combining benefit, separation, support, and two-proportion evidence.
Honest causal tree	Transformed-outcome tree with a within-training structure/ranking split.	Honest ranking fraction $= 0.5$, tree max depth from rule cap, min leaf tied to arm support and leaf floor.	Leaf effects estimated on the honest ranking split, then support and lower $\mathrm{RD}_{\mathrm{bad}}$.
Causal forest surrogate	Transformed-outcome random forest distilled into a shallow interpretable tree.	RF regressors $= 300$, RF leaf size $\max(2, 30/2)$, surrogate max depth from rule cap, surrogate min leaf $= 30$.	Surrogate CATE score, training support, and lower training-side $\mathrm{RD}_{\mathrm{bad}}$.
CURLS	Wrapper around the released CURLS implementation.	Thresholds $= 4$, negated literals enabled, max selected features $= 60$, variance weight $= 0$, minimum coverage $= 0.1$, rule length from cap.	CURLS exported order is preserved; no held-out reranking is applied.
MaxTE Subgroups	Strict R wrapper using `rpart`, pruning by minimum cross-validation error, and rule extraction.	Numeric-only feature interface, within-training structure/ranking split $= 0.5$, Rscript path fixed, rule-length cap from runner.	Ranking-split treatment-benefit estimate from the R helper, followed by the Python rule-length cap.

`-require-paired-splits`, the renderer aborts final Table 1 generation if any method–task row has fewer than 50 used artifacts or if the split-id sets differ across methods within a task.

RuAG clinical adapter. We adapt the released rule-learning component of RuAG (Zhang et al., 2025b) to the six clinical frames and archived patient splits, retaining its MCTS implementation with task-specific predicate and reward interfaces. The adapter uses eligible covariates and, for dynamic frames, fixed longitudinal mean, maximum, and last-minus-first summaries under the task's measurement policy. Numeric imputation and categorical encoding are fitted on the training partition. Qwen/Qwen3.5-9B, accessed through SiliconFlow with temperature 0, thinking disabled, and a 4096-token output budget, receives the task context, supplied prior records, and available feature catalogue to select at most 32 features. The prompt contains no patient rows. Duplicate selections are removed and excess entries are truncated in the model's returned order.

Binary features, including one-hot indicators, yield equality predicates; other numeric features receive at most three split points from univariate Gini trees fitted to the training outcome, using both $\leq$ and $>$ literals. MCTS uses a budget of 2000 rollouts, exploration weight 0.7, and conjunctions of at most three predicates on distinct features. Supported rules require at least 30 training patients, with reward $(\mathrm{ARR}_{\mathrm{train}} + 1)/2$. Only supported terminal rules actually simulated by MCTS enter the exported candidate list. They are ranked by training ARR, with ties resolved by larger minimum arm size, shorter rule length, and lexicographic rule order. The first rule is frozen before held-out replay and evaluated using the Table 1 metrics. RuAG's downstream answer-generation stage is outside this subgroup-discovery evaluation.

Metric definitions. Let $D_{d,b}^{\text{test}}$ be the held-out patient set for task frame d and split b, and let

$$S(q) = \{i \in D_{d,b}^{\text{test}} : q(x_i) = 1\}$$

be the subgroup selected by rule q. For treatment arm $t \in \{0, 1\}$,

$$\begin{aligned} n_t(q) &= \sum_{i \in S(q)} \mathbf{1}\{T_i = t\}, \\ \hat{p}_{t,\text{bad}}(q) &= \frac{\sum_{i \in S(q)} \mathbf{1}\{T_i = t\} Y_i^{\text{bad}}}{n_t(q)}. \end{aligned} \tag{26}$$

The current main-text Table 1 reports the benefit-oriented bad-outcome absolute risk reduction,

$$\begin{aligned} \widehat{\text{RD}}_{\text{bad}}(q) &= \hat{p}_{1,\text{bad}}(q) - \hat{p}_{0,\text{bad}}(q), \\ \widehat{\text{ARR}}_{\text{bad}}(q) &= -\widehat{\text{RD}}_{\text{bad}}(q) \\ &= \hat{p}_{0,\text{bad}}(q) - \hat{p}_{1,\text{bad}}(q). \end{aligned} \tag{27}$$

Higher $\widehat{\text{ARR}}_{\text{bad}}$ is better: positive values mean that the committed subgroup has a lower bad-outcome rate under the treatment arm than under the control arm. The six task columns `B-D-BL`, `B-D-TR`, `P-M-BL`, `P-M-TR`, `P-A-FPG`, and `P-A-AUC` are the mean held-out $\widehat{\text{ARR}}_{\text{bad}}$ values for the method's Best-1 rules in the corresponding clinical frame.

The three rightmost columns are aggregate diagnostics over the same Best-1 replay records. Let

$$\text{cov}(q) = \frac{|S(q)|}{|D_{d,b}^{\text{test}}|} \tag{28}$$

be held-out subgroup coverage. Population-adjusted ARR is a descriptive impact summary,

$$\text{PARR}(q) = \widehat{\text{ARR}}_{\text{bad}}(q)\, \text{cov}(q), \tag{29}$$

and the displayed `P-ARR` entry averages this quantity over the six task frames. It is included as a coverage-adjusted summary of the displayed ARR effects, not as an independent causal endpoint. Usable support is an outcome-free support-validity indicator,

$$U(q) = \mathbf{1}\left\{0.10 \le \text{cov}(q) \le 0.60 \quad \text{and} \quad \min\{n_0(q), n_1(q)\} \ge 5\right\}, \tag{30}$$

and `U-Supp.` is 100 times the mean of $U(q^{\star}_{m,d,b})$, averaged over the six task frames. This statistic penalizes both tiny subgroups and near-population rules, while requiring minimal arm-specific support. Direction consistency compares the discovery-side treatment-benefit direction with the held-out direction. If $A^{\text{train}}_{m,d,b}$ denotes the training-side benefit sign recorded for the exported Best-1 rule and $A^{\text{test}}_{m,d,b} = \widehat{\text{ARR}}_{\text{bad}}(q^{\star}_{m,d,b})$, then

$$\text{DCons}_m = 100\, \frac{\sum_{d,b} \mathbf{1}\left\{\text{sign}(A^{\text{train}}_{m,d,b}) = \text{sign}(A^{\text{test}}_{m,d,b})\right\}}{\sum_{d,b} \mathbf{1}\left\{A^{\text{train}}_{m,d,b} \text{ and } A^{\text{test}}_{m,d,b} \text{ are nonmissing}\right\}}. \tag{31}$$

This is the Best-1 D-Cons. column in Table 1. For SCENE, all 300 committed Best-1 outputs have matched direction records and preserve direction. The reporting predicate uses discovery-visible checks; held-out directions cannot filter outputs or change their ranking. The 84.9% archive-level value in Appendix B.4 uses the broader eligible Top-5 replay population and a different denominator.

Arm-specific bad-outcome rates, raw subgroup coverage, smaller-arm support, and risk-difference intervals remain in the run-level audit files but are not displayed in the compact main-text table.

Aggregation. For a scalar run-level metric z used by Table 1, the task-level entry for method m and task frame d is the mean over used split artifacts,

$$\bar{z}_{m,d} = \frac{1}{B_{m,d}} \sum_{b \in \mathcal{I}_{m,d}} z_{m,d,b}, \tag{32}$$

where $\mathcal{I}_{m,d}$ is the set of split identifiers marked `used` in the manifest and $B_{m,d} = |\mathcal{I}_{m,d}|$. The six ARR columns use $z_{m,d,b} = \widehat{\mathrm{ARR}}_{\mathrm{bad}}(q^{\star}_{m,d,b})$. The `P-ARR` and `U-Supp.` columns first form the task-level means of $\mathrm{PARR}(q^{\star}_{m,d,b})$ and $100U(q^{\star}_{m,d,b})$, respectively, and then average those task-level means across the six clinical frames so that each frame receives equal weight. `Best-1 DC` is computed from the nonmissing run-level direction labels across the six frames. No missing metric is imputed; files with missing required columns or invalid Best-1 rules are recorded in the skipped-file manifest.

Single-split Wald intervals for two proportions are still stored in the run-level artifacts for audit,

$$\widehat{\mathrm{SE}}(q) = \sqrt{\frac{\hat{p}_{1,\mathrm{bad}}(q)(1-\hat{p}_{1,\mathrm{bad}}(q))}{n_1(q)} + \frac{\hat{p}_{0,\mathrm{bad}}(q)(1-\hat{p}_{0,\mathrm{bad}}(q))}{n_0(q)}}, \tag{33}$$

but these per-rule intervals are not used to choose rules and are not part of the compact main-text Table 1.

Interpretation boundary. This implementation is intended to make the Table 1 comparison about rule discovery rather than about leakage, outcome redefinition, or test-set rule search. The shared endpoint and exclusion policy align the covariate boundary across methods; the best-1 boundary prevents a method from searching the held-out set over many exported rules; and the manifest makes the repeated-split accounting auditable. These design choices support the paper's claim at the level of exploratory, prioritized clinical subgroup hypotheses. They do not turn the resulting rules into confirmatory treatment-effect estimates, valid post-selection p-values, or clinical recommendations.

A.12 Implementation Details for Table 2

The component-ablation rows in Table 2 are implemented as a paired L1000 second-scene module ablation. The unit of analysis is one complete discovery episode for a fixed target mechanism g, one train/holdout split and search seed s, and one ablation profile a. Each episode freezes the target mechanism, the split, and the search seed before holdout evaluation, and then executes the full second-scene chain:

$$\begin{array}{c}\text{L1000 table}\\ \text{construction}\end{array} \rightarrow \text{SCENE discovery} \rightarrow \begin{array}{c}\text{holdout}\\ \text{rule replay}\end{array} \rightarrow \begin{array}{c}\text{knowledge-proposition}\\ \text{export.}\end{array}$$

The paper experiment uses three target mechanisms and ten split/search seeds, so the reporting contract for each ablation row is

$$|\mathcal{G}| \times |\mathcal{S}| = 3 \times 10 = 30$$

paired target-seed episodes. The convenience runner `run_l1000_ablation_table2.py` can execute and collect runs for a single prepared modeling table. For the manuscript table, we repeat that runner over the three target-specific modeling tables and then use `report_table2_table3_contracts.py` to enforce the full target-seed-profile matrix before aggregation. A Table 2 row is reportable only if this checker finds exactly one run for every expected target-seed cell and every reported metric is finite in all 30 cells; otherwise it writes `table2_pairing_errors.csv` or `table2_metric_errors.csv` and aborts.

L1000 table construction. The reported experiments use a fixed raw-to-modeling conversion script that constructs one row per L1000 signature instance. Each row in the resulting modeling table corresponds to one L1000 signature instance indexed by `sig_id`. Static fields describe the cell and perturbagen context, while dynamic fields summarize dose, time, pathway, and landmark-gene response programs. The default paper-facing builder uses LINCS/L1000 Level-5 COMPZ signatures and metadata resources (Lamb et al., 2006; Subramanian et al., 2017; Keenan et al., 2018; Koleti et al., 2018); pathway summaries use curated molecular-signature programs (Liberzon et al., 2015). It keeps compound perturbations for the modeling population, uses genetic perturbations as target-mechanism references, applies `tas_min=0.2`, keeps 6h/24h/96h signatures, retains the top 20 cell contexts after a minimum 200 signatures-per-cell filter, caps the modeling subset at 50,000 signatures, caps the reference pool at 5,000 signatures, and reads the GCTX matrix in column batches of 2,048.

For a target mechanism g, the builder constructs a target centroid from reference perturbation signatures when at least `min_target_ref=20` references are available. Cell-specific target centroids

are used when at least `min_cell_target_ref=3` references exist for that cell; otherwise the global target centroid is used. The continuous endpoint used by Table 2 is the configured connectivity score c_i, recorded as `connectivity_score` and related target-margin fields. The binary `outcome` field is retained for compatibility with shared rule-mining utilities, but Table 2 reports holdout connectivity contrasts rather than treating this compatibility label as the primary endpoint.

External proposal control. The One-shot LLM row uses the same target–seed pairing and replay metrics, but its proposal protocol is separate from the SCENE ablation runner. Appendix B.9 specifies its single-pass construction and paired comparison; the ablation commands and profile artifacts below describe the original SCENE component runs.

Leakage controls. Discovery excludes identifiers, split keys, target-derived connectivity fields, outcome columns, and known assay-density shortcuts from rule literals. In L1000, perturbagen-frequency features are disabled by default, and tokens such as `distil_`, `tas`, `pct_self_rank_q25`, and `dose_time_exposure` are hard-blocked. The preferred holdout split is group-aware by `pert_iname`; the split key is `sig_id` when unique and otherwise the internal row id. The default holdout fraction is 0.25. The split specification is written to `l1000_split_spec.json` and reused by the evaluator, so validation metrics cannot alter the rule search in the same episode.

Ablation matrix. All profiles share the same L1000 adapter, target construction, split policy, rule grammar, and evaluator. They differ only in which SCENE modules are enabled. The internal profile names are kept in commands and manifests, while the paper-facing labels match Table 2:

- `full`: enables all modules and is reported as **SCENE (ours)**.
- `no_top_layer`: removes the upper strategic layer while retaining lower-level search, and is reported as **w/o Upper**.
- `no_dynamic`: removes dynamic feature planning and executable dynamic feature replay, and is reported as **w/o Dynamic**.
- `no_virtual`: removes virtual feature construction and refinement, and is reported as **w/o Virtual**.
- `no_pareto`: disables Pareto multi-objective selection and keeps scalar-fitness selection, and is reported as **w/o Pareto**.
- `minimal_baseline`: removes the ablated modules jointly and is reported as **Minimal**.

Unless overridden by an ablation profile, the shared mining knobs are population size 160, 14 generations, maximum rule length 3, minimum subgroup size 60, initial feature sample size 32, support metric `transfer_support`, generalization group `pert_iname`, minimum group coverage 12, minimum group coverage ratio 0.01, and support-size weight 0.22.

MaxTE-Context adaptation. MaxTE-Context adapts MaxTE Subgroups' single-tree pipeline (Yang et al., 2026) to continuous connectivity rather than causal treatment effects. We use RPS6, MYC, and AURKB with seeds 44–53, 25% `pert_iname`-group holdouts, and a group-disjoint 50:50 structure/ranking split within discovery data. Admissible features use structure-fitted median imputation and one-hot encoding. One `rpart` regression tree (initial $cp = 0.01$, depth ≤ 3) is pruned by minimum 10-fold row-wise cross-validation error. On the ranking partition, leaf rules require at least 60 subgroup and 60 complement signatures, 12 perturbagens, 1% perturbagen-group coverage, and at most 95% sample coverage; only positive subgroup-minus-complement mean connectivity contrasts are retained, in descending order. There is no LLM, sequential covering, or ranking-feedback refitting. The available 2–5 rules per run are frozen without padding to 20 and replayed using the five metric definitions below; Gap and retrieval votes use full-discovery contrasts, and the 30 episode means are equally weighted. Metric definitions are shared, but rule counts are not matched. The positive-leaf reporting policy was revised after earlier holdout results; this is a descriptive adaptation comparison, not an independent confirmatory test.

LLM and feature-generation routes. The L1000 discovery script uses separate third-party LLM API routes for structured JSON decisions, free-form text synthesis, dynamic feature planning, and

dynamic code generation; it does not require local GPU inference. The paper configuration records the provider, model name, retry policy, timeout, and `enable_thinking` setting in `l1000_run_manifest.json`. The JSON route defaults to `Qwen/Qwen3.5-9B` (Qwen Team, 2026a) with JSON response format and thinking disabled; text synthesis uses the configured text route; dynamic planning and code generation use their dedicated route configs. CPU-side aggregation was run on a single CPU workstation (AMD Ryzen 7 7840H, 16 GB RAM). Individual target–seed–profile episodes typically require tens of minutes to a few CPU-hours after the modeling table is prepared. Exact bitwise replay of live LLM calls requires the archived prompts and returned artifacts; the numerical Table 2 aggregation is based on the saved run artifacts, not on re-querying the LLM.

Table 14: Artifact contract for Table 2. These files make the target-seed pairing, split boundary, ablation state, and metric source auditable.

Artifact	Role	Required audit fields
`l1000_build_summary.json`	Modeling-table provenance.	Raw L1000 path, target perturbagen, reference counts, filters, row counts, output paths.
`l1000_run_manifest.json`	Single-run manifest.	Target mechanism, ablation profile, seed, enabled components, split-spec path, train/holdout sizes, LLM configs, blocked feature tokens.
`l1000_split_spec.json`	Frozen discovery/holdout split.	Split key, train/test keys, split group column, holdout fraction, random seed.
`l1000_agentic_results.csv`	Discovery-side ranked rules.	Rule text, fitness, Pareto rank, discovery contrast, support, transfer support, group coverage, p-value, generation.
`test_rule_eval.csv` and `l1000_quality_eval.json`	Holdout rule replay.	Rule validity, subgroup size, support ratio, ΔConn, PosRate90, p-values, rank stability, generalization gap.
`l1000_table2_metrics.json`	Run-level Table 2 source.	Compact metric set consumed by the Table 2 collector.
`l1000_knowledge_propositions.json`	Downstream knowledge export.	Selected validated rules, target mechanism, support/effect evidence, feature tokens, source labels.

Holdout evaluation. Rules are ranked before holdout replay by discovery-side order: Pareto rank when enabled, fitness, training-side connectivity contrast, support, p-value, and generation. The evaluator replays the top $K = 20$ rules on the frozen holdout table; the quality summary also uses $K = 20$ unless the run manifest records an override. Virtual features are fit on the discovery-training side and replayed on holdout. Dynamic features are exported as executable plans and replayed before validation. If a rule depends on a virtual or dynamic feature that cannot be replayed on holdout, that rule is marked invalid and contributes to the valid-rule-rate denominator.

Let $D^{\text{test}}_{g,s}$ be the frozen holdout set for target g and seed s, and let c_i be the configured connectivity score. For a rule q,

$$S(q) = \{i \in D^{\text{test}}_{g,s} : q(x_i) = 1\}, \qquad B(q) = D^{\text{test}}_{g,s} \setminus S(q).$$

The mean holdout connectivity contrast is

$$\Delta\text{Conn}_{\text{test}}(q) = \frac{1}{|S(q)|} \sum_{i \in S(q)} c_i - \frac{1}{|B(q)|} \sum_{i \in B(q)} c_i.$$

The median version replaces the two means by medians. The strong-connectivity threshold τ is fixed inside the evaluator. In `auto` mode, the evaluator uses a CLUE-like $\tau = 90$ threshold when the score scale supports it and otherwise uses the holdout 90th percentile. Thus

$$\text{PosRate90}(q) = |S(q)|^{-1} \sum_{i \in S(q)} \mathbf{1}\{c_i \geq \tau\},$$

and $\Delta\mathrm{PosRate90}(q)$ subtracts the corresponding background rate over $B(q)$.

Metrics and aggregation. For one component-ablation run, Table 2 summarizes a valid top-K replay subset $\mathcal{Q}_{a,g,s}$ for ablation profile a, target g, and seed s. Quality metrics are computed from valid replayed rules, with one support focus used by the evaluator. If at least $\max(5, \lceil K/2 \rceil)$ valid rules have `support_ratio_test >= 0.02`, the mean contrast, PosRate90, p-value rate, and gap diagnostics are computed on this support-focused subset and the run records `quality_focus_mode_topk=support_ge_0.02`; otherwise the evaluator uses all valid top-K rules and records `quality_focus_mode_topk=all`. In both cases, `Valid Rule Rate` uses the raw top-K replayed rules as its denominator, including rules with parsing errors or empty subgroups. The focus mode and focus rule count are preserved in `l1000_table2_metrics.json` and in the reporting-contract manifest.

The `Conn.` column is the run-level mean of the held-out connectivity contrast,

$$\mathrm{Conn}_{a,g,s} = \frac{1}{|\mathcal{Q}_{a,g,s}|} \sum_{q \in \mathcal{Q}_{a,g,s}} \Delta\mathrm{Conn}_{\mathrm{test}}(q), \tag{34}$$

where larger values indicate stronger target-directed connectivity inside the rule-covered signatures than in the held-out background. The `Strong-Pos.` column is 100 times the corresponding mean $\mathrm{PosRate90}(q)$ over $\mathcal{Q}_{a,g,s}$, so it measures the fraction of rule-covered signatures falling into the strong-response tail. The significance-rate diagnostic uses effective holdout $p < 0.05$, with permutation p-values automatically refined from 1,000 to at most 5,000 permutations when floor saturation is detected. These p-values are retained in the audit files but are post-selection diagnostics, not confirmatory inference for adaptively searched rules.

Target retrieval is evaluated by a rule-ensemble score on holdout signatures. Each top discovery rule votes for matching signatures with positive discovery-side contrast when available, otherwise a unit vote. The preferred positive label is `connectivity_target_margin > 0`; if that label is unavailable, the evaluator falls back to the strong-connectivity threshold. `AUPRC` is the area under the precision–recall curve of this ensemble score, following standard PR evaluation conventions for imbalanced binary retrieval (Davis & Goadrich, 2006; Saito & Rehmsmeier, 2015); AUROC, Hit@K, recall@K, precision@K, and target-rank fields are also stored in the audit files following PR/ROC conventions (Fawcett, 2006). When perturbagen names and a target perturbagen are available, perturbagen-level median scores are used for target rank and Hit@K.

The current compact table reports `U-Supp.` rather than raw support ratio. For a replayed L1000 rule,

$$U_{\mathrm{L1000}}(q) = \mathbf{1}\left\{0.10 \le \frac{|S(q)|}{|D_{g,s}^{\mathrm{test}}|} \le 0.60 \quad \text{and} \quad |S(q)| \ge 5\right\}, \tag{35}$$

and `U-Supp.` is $100|\mathcal{Q}_{a,g,s}|^{-1} \sum_{q \in \mathcal{Q}_{a,g,s}} U_{\mathrm{L1000}}(q)$, averaged across episodes. This makes the reported support diagnostic a usability rate rather than a monotone reward for larger subgroups. The `Gap` column is the normalized train–holdout contrast gap,

$$\mathrm{Gap}_{a,g,s} = \frac{\left|\overline{\Delta\mathrm{Conn}}_{\mathrm{train},a,g,s} - \overline{\Delta\mathrm{Conn}}_{\mathrm{test},a,g,s}\right|}{\max\{10^{-8}, |\overline{\Delta\mathrm{Conn}}_{\mathrm{train},a,g,s}|\}}, \tag{36}$$

so smaller values indicate better transfer from discovery to holdout.

For scalar metric z, the Table 2 entry for ablation profile a is

$$\bar{z}_a = \frac{1}{30} \sum_{g \in \mathcal{G}} \sum_{s \in \mathcal{S}} z_{a,g,s}.$$

The reporting artifacts store the mean and empirical standard deviation over the same 30 target-seed episodes; the compact main-text table reports the means only. No missing numeric metric is imputed. A run is nonreportable if discovery produces no rule export, if holdout replay fails, if required features cannot be materialized, or if `l1000_table2_metrics.json` is missing.

Interpretation boundary. Table 2 evaluates whether SCENE modules improve replayable L1000 mechanism-connectivity discovery under fixed targets and splits. The output should be interpreted as

context-bounded perturbational association evidence: a rule identifies cell, perturbagen, dose/time, or signature contexts with stronger held-out target connectivity. It is not a clinical recommendation, an externally validated biological mechanism, or a causal compound-effect estimate.

Reproduction command. The paper-facing execution uses `run_l1000_ablation_table2.py`; the script default contains a convenience preset, while the paper run overrides it with ten seeds and repeats the runner for the three target mechanisms. Final manuscript aggregation is performed by `report_table2_table3_contracts.py`, which checks exact one-run-per-cell coverage and per-metric completeness for the three-target by ten-seed matrix before writing `table2_paired_run_manifest.csv` and `table2_paired_summary.csv`. A reproducible shell form is:

Reproduction pipeline

```
$ProjectRoot = "<absolute path to the project root containing analysis/>"
$PythonExe = "<absolute path to the Python executable>"
$OutRoot = "<absolute path to the Table 2 output root>"
$AnalysisDir = Join-Path $ProjectRoot "analysis"
$Targets = @("<target_1>", "<target_2>", "<target_3>")
$Seeds = "44,45,46,47,48,49,50,51,52,53"
$Profiles = "full,no_top_layer,no_dynamic,no_virtual,no_pareto,
    minimal_baseline"

foreach ($target in $Targets) {
  $tableDir = Join-Path (Join-Path $OutRoot "tables") $target
  & $PythonExe (Join-Path $AnalysisDir "prepare_l1000_tables.py") `
    --target-pert-iname $target `
    --out-dir $tableDir

  $runRoot = Join-Path (Join-Path $OutRoot "runs") $target
  & $PythonExe (Join-Path $AnalysisDir "run_l1000_ablation_table2.py") `
    --modeling-csv (Join-Path $tableDir "l1000_modeling_table.csv") `
    --result-root $runRoot `
    --study-config paper_table2_v1 `
    --profiles $Profiles `
    --seeds $Seeds `
    --run `
    --generations 14 `
    --population-size 160 `
    --min-subgroup-size 60 `
    --eval-top-k 20 `
    --eval-quality-top-k 20 `
    --out-dir (Join-Path $runRoot "summary")
}
```

A.13 Implementation Details for Table 3

Table 3 evaluates whether contextualized knowledge discovered by SCENE can be reused as auxiliary knowledge for downstream few-shot tabular classification under an in-context learning protocol (Brown et al., 2020). This is a post-discovery utility experiment: Table 1 and Table 2 produce rules and knowledge propositions; Table 3 converts them into compact knowledge cards and asks whether they improve few-shot predictions under the same row serialization, exemplar budget, split, and LLM decoding policy. No Table 3 prediction, prompt, or metric is fed back into the upstream discovery runs.

Scenes and labels. The core runner is `run_knowledge_augmented_fewshot.py`, with task context supplied by `knowledge_base.py`. The active scene is selected by `CODE_CONFIG["active_scene"]` or by the corresponding command-line arguments. Clinical tasks use patient-level records. In the paper-facing treatment-benefit mode, the positive class is evidence consistent with a positive treatment-over-control signal: a treatment-arm record with a good outcome or a control-arm record with a bad outcome. This is a row-level downstream classification proxy for treatment-benefit evidence, not an individual treatment-effect estimate.

For L1000, the row is a signature instance. The preferred paper label is an external target-mechanism label when available. When the L1000 run is configured with `l1000_label_mode=target_margin`, the positive class is the target-mechanism proxy `connectivity_target_margin > 0`. If `strong_connectivity` is used, it is reported as a secondary stress test. Label columns, target-derived connectivity fields, `outcome`, `sig_id`, `pert_id`, `distil_id`, internal row ids, SMILES/InChI keys, and manifest hard-block tokens are excluded from row serialization and from rule-card literals unless an explicit stress-test mode enables them. Biological context fields such as `cell_id`, dose/time fields, and `pert_iname` may remain in the serialized row; when `pert_iname` is available, the downstream split is group-aware by `pert_iname` and the split manifest records train-test group overlap.

Split and exemplar protocol. Each run fixes a scene, one outer split, one few-shot exemplar set, one prompt condition, and one deterministic LLM decoding setting. The default train/validation/test fractions are 0.6/0.2/0.2 with split seed 44, unless a source split is supplied. Clinical runs can bind the final test partition to upstream source test identifiers. L1000 runs bind to the selected source `l1000_split_spec.json` when `-use-source-test-ids` is enabled; with `-require-group-split`, train/validation/test splitting is group-aware by `pert_iname` and the split manifest records train-test group overlap. Few-shot exemplars are sampled only from the train partition, while knowledge-card selection is scored only on the validation partition. The final test partition is used only for Table 3 prediction and metric computation.

Table 3 uses three prompt conditions and ten exemplar seeds per condition in the paper configuration. The checked source of truth is the command line and the resulting `run_config.json`; the smaller seed list in `CODE_CONFIG` is a convenience/debug default and is not used for the manuscript table. C0 is the ordinary few-shot baseline with row serialization and labeled exemplars. C1 adds generic scene/task context to the same few-shot protocol. C2 uses the SCENE discovered-card protocol: task-specific scene context, task knowledge context when available, row-conditioned discovered rule cards, and the same few-shot exemplars. The runner writes these same condition names in all metric files; `report_table2_table3_contracts.py` checks that all three conditions exist for each scene/shot/source directory, checks the exact exemplar seed set and duplicate-free condition-seed rows, checks L1000 split/leakage invariants when `pert_iname` is serialized, and writes the manuscript-facing `table3_mapped_metrics_summary.csv` and `table3_mapped_metrics_agg.csv`.

Knowledge-card construction. Candidate cards are loaded from the upstream rule artifacts of the corresponding source run. The loader parses rule expressions, canonicalizes semantic duplicates, removes rules using prohibited or blacklisted fields, and scores the remaining rules on the validation split only. For ordinary binary classification, the validation card effect is

$$\begin{aligned}\Delta_{\mathrm{val}}(q) = \ & \Pr(Y = 1 \mid q(X) = 1, D^{\mathrm{val}}) \\ & - \Pr(Y = 1 \mid D^{\mathrm{val}}),\end{aligned}$$

and the ranking score is $\Delta_{\mathrm{val}}(q)\sqrt{n_{\mathrm{val}}(q)}$. For clinical treatment-benefit mode, the validation effect is the within-rule treatment-minus-control good-outcome contrast,

$$\begin{aligned}\Delta_{\mathrm{val}}^{\mathrm{benefit}}(q) = \ & \Pr(G = 1 \mid q(X) = 1, T = 1, D^{\mathrm{val}}) \\ & - \Pr(G = 1 \mid q(X) = 1, T = 0, D^{\mathrm{val}}),\end{aligned}$$

and support is $\min(n_{\mathrm{treat}}, n_{\mathrm{ctrl}})$ inside the rule. The paper discovered-card pool keeps validation-directionally helpful rules with `min_card_val_effect_delta=0.0`, applies the minimum validation support threshold, and selects the top `top_k_cards=5` cards unless a run manifest records an override.

Each selected card stores a card id, rule condition, consequence text, validation effect, validation support, and source-run evidence. Cards are treated as soft priors, not deterministic labels. With `row_conditioned_cards=True`, a query prompt receives only selected cards whose rule condition matches the query row. If no selected card matches, the prompt explicitly states that no selected rule card matches the current row. For L1000, `use_l1000_knowledge_json_cards=False` is the default in this few-shot runner to avoid reusing cards selected by the same upstream holdout when that holdout is also bound as the final downstream test split.

Agent and compute resources. Table 3 uses third-party LLM API inference for the few-shot prediction calls in all prompt conditions. C0, C1, and C2 share the same model route, decoding policy, exemplar budget, and parsing rules; only the supplied knowledge context differs. The CPU-side aggregation was run on a single CPU workstation (AMD Ryzen 7 7840H, 16 GB RAM). No local GPU is required unless one chooses to replace the API route with local LLM inference. The CPU-side reporting work is lightweight once prompts and responses are cached; end-to-end wall-clock time is dominated by provider-dependent API latency for the 2 scenes, 3 prompt conditions, and 10 exemplar seeds.

Table 15: Artifact contract for Table 3. These outputs make the few-shot split, card selection, prompts, model outputs, and parsed predictions auditable.

Artifact	Role	Required audit fields
`run_config.json`	Resolved downstream configuration.	Scene, task id, source run directory, data/rule/knowledge paths, label source, conditions, shots, exemplar seeds, row features, LLM fields.
`split_indices.json`	Downstream split manifest.	Train/validation/test indices, split seed, group column, source-test binding, fixed-split information.
`candidate_rules_scored.csv`	Validation-side card ranking.	Rule, semantic key, validation support, validation effect, validation score, leakage-filter status.
`selected_discovered_cards.json`	Paper C2 card source.	Card ids, rules, validation effects, supports, card group, source evidence.
`prompts/*.json`	Full prompt replay record.	System prompt, user prompt, exemplar ids, active card ids, raw output, repair output, parsed answer, gold label.
`fewshot_predictions.csv`	Row-level predictions.	Condition, shot, seed, query row, true label, parsed prediction, parsed probability, active cards, parse-failure flag.
`fewshot_metrics_summary.csv` and `fewshot_metrics_agg.csv`	Table 3 metric sources.	Per-condition/per-seed metrics and condition-level mean/std summaries.
`table3_mapped_metrics_summary.csv` and `table3_mapped_metrics_agg.csv`	Final paper-label reporting files.	C0/C1/C2 completeness, exact seed-set and duplicate checks, L1000 leakage checks, final Table 3 aggregate metrics.

Prompt and inference protocol. Rows and exemplars are serialized as feature-value statements using the same selected row feature list for all conditions in a scene. The selector first keeps configured canonical features, then scene-priority features, then features used by selected cards, and finally fills remaining slots up to `max_row_features=24`; blacklist tokens such as follow-up, recurrence, death, outcome, and event fields are applied before serialization. Few-shot sampling is class-balanced; in clinical `label_group` mode it also balances treatment arm when possible to reduce treatment-label shortcut prompts.

The paper run uses real LLM inference, `temperature=0.0`, thinking disabled when supported by the provider, prompt saving enabled, and `llm_parse_fallback=error`. The configured model and provider are recorded in `run_config.json`; our reporting command fixes `Qwen/Qwen3.5-9B` (Qwen Team, 2026a) for the few-shot runner unless a run manifest explicitly records another paper-approved model. Model outputs are parsed as yes/no answers from `<answer>` tags and task-specific positive/negative hints. If the first parse fails, the script performs one repair call that asks only for `<answer>yes</answer>` or `<answer>no</answer>`. Under the paper fail-closed policy, unresolved parse failures abort the run rather than being replaced by mock or majority predictions.

Metrics and aggregation. For condition c, shot count k, exemplar seed e, and final test set D^{test}, the script records parsed binary predictions $\hat{y}_i$ and parsed binary scores $\hat{p}_i$. Table 3 reports three standard binary-classification metrics. AUPRC is average precision, i.e. the area under the

precision–recall curve induced by the parsed scores (Davis & Goadrich, 2006; Saito & Rehmsmeier, 2015),

$$\mathrm{AUPRC}_{c,k,e} = \mathrm{AP}\left(\{(y_i, \hat{p}_i) : i \in D^{\mathrm{test}}\}\right).$$

MacroF1 is the unweighted mean of the class-specific F1 scores,

$$\mathrm{MacroF1}_{c,k,e} = \frac{1}{2}\left(\mathrm{F1}_{0,c,k,e} + \mathrm{F1}_{1,c,k,e}\right),$$

where each class is treated once as the positive class. Accuracy is

$$\mathrm{ACC}_{c,k,e} = \frac{1}{|D^{\mathrm{test}}|} \sum_{i \in D^{\mathrm{test}}} \mathbf{1}\{\hat{y}_i = y_i\}.$$

Because the prompted model is constrained to a yes/no answer, AUPRC has limited score resolution and is interpreted together with these thresholded metrics. Additional audit fields include AUROC, balanced accuracy, majority-class accuracy, accuracy minus majority accuracy, sensitivity, specificity, precision, test positive rate, predicted positive rate, and parse-failure count (Fawcett, 2006). For scalar metric z,

$$\bar{z}_{c,k} = \frac{1}{10} \sum_{e=1}^{10} z_{c,k,e},$$

and `fewshot_metrics_agg.csv` reports the empirical standard deviation across the ten exemplar seeds. The compact main-text Table 3 displays the means only. The final Table 3 renderer should consume `table3_mapped_metrics_agg.csv`, not the runner's intermediate aggregate, so that condition completeness, seed identity, and leakage checks are always applied.

Failure handling and interpretation. A Table 3 run is nonreportable if the label has a single class, source split binding fails, the test split is empty, candidate rules cannot be parsed, all candidates are removed by leakage filters, no discovered card survives validation scoring for paper C2, the required real LLM is unavailable, or an answer cannot be parsed under the fail-closed policy. If `max_test_samples` is used to control LLM cost, the subsampling mode, sampled class prevalence, and sampled row count are written to the metric files and run config. Missing cards are not replaced with generic biomedical text. Table 3 therefore measures the downstream prompt utility of SCENE-derived contextualized knowledge, not the independent biological or clinical truth of every card and not a causal validation of the upstream rules.

Reproduction command. The paper run invokes the three manuscript conditions directly:

Reproduction pipeline

```
$ProjectRoot = "<absolute path to the project root containing analysis/>"
$PythonExe = "<absolute path to the Python executable>"
$OutRoot = "<absolute path to the Table 3 output root>"
$env:PYTHONPATH = $ProjectRoot
$env:SILICONFLOW_API_KEY = "<provider API key>"
$env:SILICONFLOW_BASE_URL = "<OpenAI-compatible base URL>"

& $PythonExe -m analysis.run_knowledge_augmented_fewshot `
  --scene clinical `
  --task-id <clinical_task_id> `
  --source-run-dir "<absolute path to the Table 1 source run directory>"
    `
  --source-scenario both `
  --source-run-id run_01 `
  --conditions C0,C1,C2 `
  --shots 4 `
  --seeds 0,1,2,3,4,5,6,7,8,9 `
  --top-k-cards 5 `
  --min-card-val-effect-delta 0.0 `
  --row-conditioned-cards `
  --fewshot-sampling-mode label_group `
  --temperature 0.0 `
```

```
  --llm-model Qwen/Qwen3.5-9B `
  --no-llm-enable-thinking `
  --save-prompts `
  --strict-source-binding `
  --require-group-split `
  --require-real-llm `
  --llm-parse-fallback error `
  --out-dir (Join-Path $OutRoot "clinical")

& $PythonExe -m analysis.run_knowledge_augmented_fewshot `
  --scene l1000 `
  --task-id <l1000_task_id> `
  --source-run-dir "<absolute path to the Table 2 full-profile source run
     directory>" `
  --use-source-test-ids `
  --l1000-label-mode target_margin `
  --conditions C0,C1,C2 `
  --shots 4 `
  --seeds 0,1,2,3,4,5,6,7,8,9 `
  --top-k-cards 5 `
  --min-card-val-effect-delta 0.0 `
  --row-conditioned-cards `
  --temperature 0.0 `
  --llm-model Qwen/Qwen3.5-9B `
  --no-llm-enable-thinking `
  --save-prompts `
  --strict-source-binding `
  --require-group-split `
  --require-real-llm `
  --llm-parse-fallback error `
  --out-dir (Join-Path $OutRoot "l1000")
```

A.14 LLM Backend Runtime Protocol

The backend-runtime comparison changes only the third-party LLM API backend while keeping the discovery scenario, split, seed, schema, population size, generation budget, support floor, and reporting predicate fixed. Total runtime is measured as wall-clock time for the full frozen discovery run, including API calls, parsing, controller validation, evolutionary scoring, and logging. Generation latency is the mean wall-clock time per evolutionary generation within that run. Because all backends are accessed through third-party APIs, these measurements are indicative runtime costs rather than hardware-normalized throughput; provider load and network conditions can change absolute values. The sweep is reported as a runtime sensitivity analysis and is not used to select rules or revise the main benchmark protocols. Figure 7 fixes the P-M-TR task, split, seed, schema, controller settings, and discovery budget, and changes only the third-party API backend. Held-out benefit is computed after discovery as bad-outcome ARR in percentage points, averaged over the top-10 discovery-ranked rules exported by each run; held-out outcomes are not used to rank or select these rules. Runtime includes API calls, parsing, controller validation, evolutionary scoring, and logging, and should be interpreted as indicative API wall-clock cost rather than hardware-normalized throughput.

B Additional Experiments and Analyses

This section provides implementation details and post-hoc evidence audits for SCENE. The post-hoc audits reuse completed clinical and L1000 outputs without additional LLM prompting or rule discovery, and no analysis involves holdout-driven model selection. The goal is to make the empirical evidence more legible along axes that are central to SCENE: effect–support geometry, contextualization of reported rules, L1000 inspectability, held-out replay reliability, semantic provenance, component behavior, generated-feature gating, grounding traces, planner consistency, report quality, the contribution of knowledge guidance, and sensitivity to explicit prior records. These audits should therefore be read as descriptive robustness and interpretation analyses rather than as new model-selection procedures.

For the clinical analyses, we use a benefit-oriented sign convention. The source summaries report $\mathrm{RD}(\mathrm{Bad})$ as treatment bad-outcome rate minus control bad-outcome rate. We therefore define

$$\begin{aligned}\mathrm{Benefit}(R) = &\ \Pr(Y_{\mathrm{bad}} = 1 \mid R, T = 0) \\ &- \Pr(Y_{\mathrm{bad}} = 1 \mid R, T = 1) \\ = &\ -\mathrm{RD}(\mathrm{Bad}).\end{aligned} \tag{37}$$

Larger values indicate a larger reduction in bad outcomes under treatment. Unless otherwise stated, the clinical support axis denotes subgroup support, and the rule-level reliability analysis uses the smaller treatment-arm count inside the rule as the held-out support measure. This convention keeps the appendix aligned with the multi-objective evidence view used by SCENE: rules should have both a favorable scenario effect and enough support to be interpretable.

B.1 Clinical Effect–Support–Uncertainty Geometry

The main clinical comparison table reports selected subgroup summaries, but a table alone does not show how treatment benefit is positioned relative to the support-validity criterion used by the primary benchmark. Raw support is difficult to interpret as a monotone objective: a rule covering nearly everyone can be uninformative, while a very small rule can be unstable. We therefore visualize the clinical comparison with the same usable-support diagnostic reported in Table 1, and then complements it with uncertainty and variability diagnostics.

For Figure 8, we use the method-level *usable support* column reported in Table 1. Usable support measures how often a method's committed Best-1 rule has enough held-out coverage to be evaluated as a subgroup without degenerating into either a tiny fragment or a near-population rule. It is computed by first forming the task-level usability rate

$$u_{t,m} = 100 \cdot \frac{1}{|\mathcal{R}_{t,m}|} \sum_{r \in \mathcal{R}_{t,m}} \mathbb{I}\big\{10\% \le s_r \le 60\%, \\ \min(n_{r,\mathrm{treat}}, n_{r,\mathrm{control}}) \ge 5\big\},$$

where $\mathcal{R}_{t,m}$ is the set of Best-1 records available for method m in clinical frame t, s_r is the held-out subgroup coverage percentage for record r, and $n_{r,\mathrm{treat}}$ and $n_{r,\mathrm{control}}$ are the held-out treated and control counts inside the subgroup. The plotted support coordinate is then

$$U_m = \frac{1}{6} \sum_{t=1}^{6} u_{t,m},$$

which is the U-Supp. entry in Table 1. This binary usability flag penalizes both very small subgroups, which can be unstable, and near-population rules, which are weak subgroup discoveries, while also requiring minimal arm-specific support. The statistic is outcome-free: it does not use treatment effects or confidence intervals, and it asks only whether the committed Best-1 rule has a support profile suitable for held-out subgroup evaluation.

Across the 48 primary method–frame summaries used in this audit, the plotted benefit values reproduce the Table 1 ARR entries. SCENE has a median benefit effect of 0.269, compared with 0.055 for the non-SCENE baselines. On the Table 1 U-Supp. axis, SCENE reaches 84.7%, whereas the strongest non-SCENE baseline reaches 70.0%. At the per-frame level, SCENE has the largest benefit-effect point estimate in all six clinical frames. The correct interpretation is therefore not that SCENE always maximizes raw subgroup size, but that its high-benefit rules also satisfy the support-validity criterion used by the primary comparison.

Figure 8 places SCENE and the clinical baselines into this benefit–usable-support landscape. Because the x-axis is the method-level Table 1 U-Supp. diagnostic, each method keeps the same support coordinate across panels, while the y-coordinate changes with the clinical frame's ARR. SCENE lies in the upper-right region across the six panels: it combines the largest frame-specific benefit values with the highest Table 1 usable support. This pattern is consistent with the role of SCENE's lower-level multi-objective search: it should not simply maximize subgroup size, nor should it collapse to the smallest high-effect subgroup.

The clinical comparison in Table 1 reports Best-1 subgroup summaries for each method and task. The table is designed to make the primary comparison compact, but it does not directly show whether

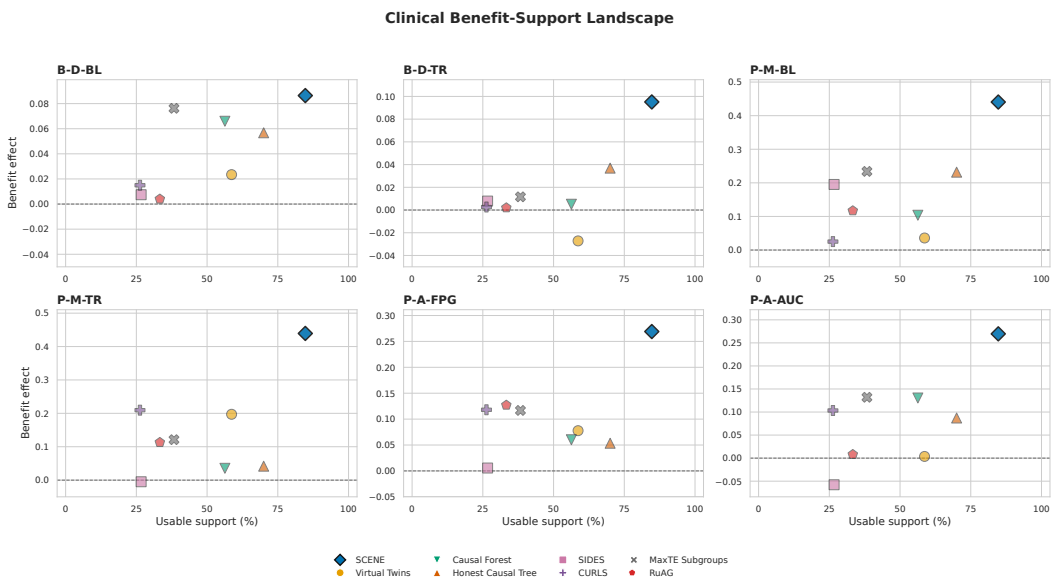


Figure 8: Clinical benefit–support landscape. Each panel shows one clinical frame. The x-axis is usable-support value from Table 1, the y-axis is the corresponding benefit effect for that frame. Marker shape and color denote the method. SCENE is consistently separated on benefit while also retaining the highest usable support.

a high benefit estimate is accompanied by unstable behavior across repeated Best-1 records. We therefore add a complementary diagnostic that removes confidence intervals from the visualization and instead summarizes empirical variability of the Best-1 benefit effect. This audit is descriptive rather than inferential: it is not used for model selection, and it is not intended to replace the patient-level confidence intervals in the main quantitative comparison. Its purpose is to check whether the favorable clinical effect estimates of SCENE are accompanied by excessive run-to-run variability.

For this analysis, the benefit effect is the bad-outcome reduction,

$$\Delta = \mathrm{ControlBadRate} - \mathrm{TreatmentBadRate}.$$

Larger values are better. The input records are aligned with the Table 1 reporting protocol: when a method exports a ranked rule list, only the prioritized rule is used as its Best-1 rule; archived replay-audit records contribute one Best-1 replay record per row. For each method and clinical frame, we first compute a task-level center and a task-level variability statistic over these Best-1 records. We then average those task-level summaries across the six Table 1 clinical frames. The resulting figures therefore show one point per method, not per-run points, and do not encode or reveal the number of available records for any method.

We use three variability definitions. The first uses the mean benefit as the center and the standard deviation (SD) as the variability statistic. The second uses the median benefit as the center and the median absolute deviation (MAD) around that median as a robust variability statistic. The third again uses the median benefit as the center but uses the interquartile range (IQR) as the variability statistic. In each plot, the horizontal axis is the benefit center, so moving right is better. The vertical axis is the empirical variability estimate, so moving upward is better because the axis is drawn with lower variability at the top. The upper-right region is therefore the favorable region: high benefit and low variability.

Figure 9 gives the most conventional variability view. SCENE has the largest mean benefit effect (0.267). The same figure also shows that SCENE does not obtain this larger benefit by becoming more variable. Its average task-level SD is 0.097, lower than every baseline in this audit. The closest baselines on variability are SIDES (0.140), Honest Causal Tree (0.155), and MaxTE Subgroups (0.165), but all three are substantially left of SCENE on the benefit axis. Conversely, CURLS and Causal Forest are not only lower-benefit but also more variable under this summary. Thus the

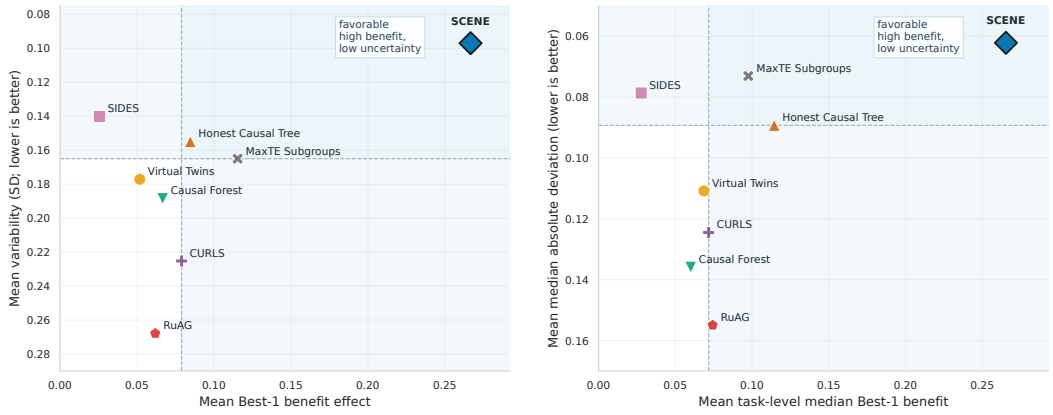


Figure 9: Clinical benefit–variability profiles. (Left) Mean benefit with task-level SD. (Right) Median benefit with task-level MAD. In both panels, moving right/up indicates higher benefit and lower empirical variability.

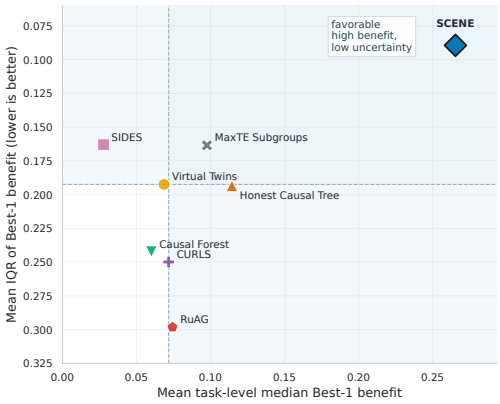
mean-SD view supports the main table's interpretation: SCENE is not merely selecting a high-effect subgroup in a noisy manner; it is separated from the baselines in the favorable direction on both axes.

The median-MAD view in Figure 9 asks whether the conclusion is driven by a small number of extreme Best-1 records. The answer is no. SCENE again has the largest center statistic, with a mean task-level median benefit of 0.265. Its average MAD is 0.062, which is also the smallest among all methods. The strongest baseline after applying the same robust summary is Honest Causal Tree by the stability-adjusted read-out, but its median benefit is only 0.114 and its MAD is 0.089. MaxTE Subgroups is close to Honest Causal Tree under this robust view, with median benefit 0.098 and MAD 0.073, but it remains far below SCENE in benefit. This is important because MAD is less sensitive to outlying records than SD. The fact that SCENE remains in the favorable upper-right region under MAD suggests that the result is not simply a mean-based artifact.

The compact audit table in Figure 10 uses the same task-level scorecard as the benefit–support landscape. The raw *Benefit effect* is the bad-outcome reduction in percentage points. The *Effect/CI signal* is the benefit effect divided by the width of the corresponding confidence interval, so it rewards effects that are large relative to their uncertainty band. The *Benefit score* and *Signal score* are task-wise min–max normalized versions of the benefit effect and Effect/CI signal, respectively. The *Composite score* is a benefit-weighted scorecard read-out, computed as 0.50 times the Benefit score plus 0.25 times the normalized support score and 0.25 times the Signal score. These score rows are therefore relative geometry diagnostics within each clinical frame, not new clinical endpoints.

Figure 10 uses IQR to focus on the middle half of the Best-1 records. This gives a second robust dispersion check that is different from MAD. The pattern remains stable. SCENE has a mean task-level median benefit of 0.265 and an average IQR of 0.089. The best baseline median benefit remains much smaller, and the baseline IQR values are all larger: SIDES and MaxTE Subgroups are around 0.163–0.164, Honest Causal Tree and Virtual Twins are around 0.193, and Causal Forest and CURLS are around 0.242–0.250. The table next to the IQR panel adds a complementary task-averaged audit from the effect–support–uncertainty scorecard. SCENE has the highest average benefit effect, effect-to-CI signal, normalized benefit score, normalized signal score, and benefit-weighted composite score across the six clinical frames, with consistent margins over the strongest baseline for each read-out. Thus the IQR view and the compact table make distinct points: the figure audits empirical dispersion, while the table summarizes method-level separation on the clinical geometry scorecard without reusing the variability axes.

Table 16 summarizes the same observation numerically. If one subtracts the variability estimate from the corresponding benefit center, SCENE remains the top method under all three non-CI diagnostics. The margin is largest in the mean-SD and median-IQR views, where all baseline adjusted values are negative. In the median-MAD view, the best baselines obtain small positive adjusted values, but they



Task-averaged geometry audit

Read-out	SCENE	Best base.	Margin
Benefit effect	26.7 pp	11.6 pp	+15.1 pp
Effect/CI signal	0.234	0.127	+0.107
Benefit score	1.000	0.430	+0.570
Signal score	0.904	0.541	+0.363
Composite score	0.976	0.520	+0.309

Figure 10: Robust clinical geometry and aggregate read-outs. (Left) Median benefit with task-level IQR. (Right) Method-level averages over the six clinical frames; the margin compares SCENE with the strongest baseline for each read-out.

Table 16: Summary of non-CI benefit–variability diagnostics. The values are method-level averages over the six Table 1 clinical frames. "Adjusted" is center minus variability and is reported only as a numerical read-out; it is not used as an additional figure axis.

Diagnostic	SCENE Center	SCENE variability	SCENE adjusted	Best baseline adjusted
Mean–SD	0.267	0.097	0.170	-0.053
Median–MAD	0.265	0.062	0.203	0.025
Median–IQR	0.265	0.089	0.176	-0.066

remain far below SCENE. This makes the qualitative conclusion robust to the choice of variability statistic.

These diagnostics should be interpreted carefully. SD, MAD, and IQR measure empirical variability across archived Best-1 records, not uncertainty in the formal statistical sense and not a replacement for held-out clinical validation. They are useful here because they avoid relying on potentially inconsistent CI endpoints from heterogeneous baseline output files and because they do not expose the number of replay records for each method. The consistent placement of SCENE across mean-SD, median-MAD, and median-IQR views provides a simple robustness check for the main clinical table: the method's advantage is not limited to a single confidence-interval rendering or a single dispersion statistic. Instead, SCENE remains the method with the highest clinical benefit and the lowest empirical variability across all three descriptive views.

B.2 Clinical Contextualization Advantage over Prior Rule Vocabularies

Table 1 evaluates whether each method's exported rule improves the held-out clinical endpoint. The present audit asks a different question: when a method commits to its top-ranked rules, do the induced subgroups also concentrate on clinically meaningful, task-relevant axes rather than arbitrary threshold combinations? The prior-method reference distributions in this audit pool the six original data-only baselines. For each clinical task frame, we define three non-outcome clinical criteria before plotting. The breast-cancer frames use standard prognostic axes from the trial setting: poor histologic grade, estrogen receptor negativity, and progesterone receptor negativity. The PCOS frames use endpoint-matched endocrine, glycemic, lipid, and insulin-resistance axes. For metformin-response tasks these include hirsutism/gonadotropin phenotype and insulin-exposure markers; for the fasting-glucose acupuncture task these include fasting-glucose, triglyceride, and C-peptide criteria. For the OGTT AUC acupuncture task, we intentionally reuse the same three metabolic-risk criteria shown in the main-text clinical-risk audit: central adiposity (waist $\geq$ 80 cm), insulin resistance (HOMA-IR $\geq$ 2.5), and elevated triglycerides (TG $\geq$ 1.7 mmol/L). These criteria are used only for post-hoc contextualization auditing and are never used to select or revise the rules.

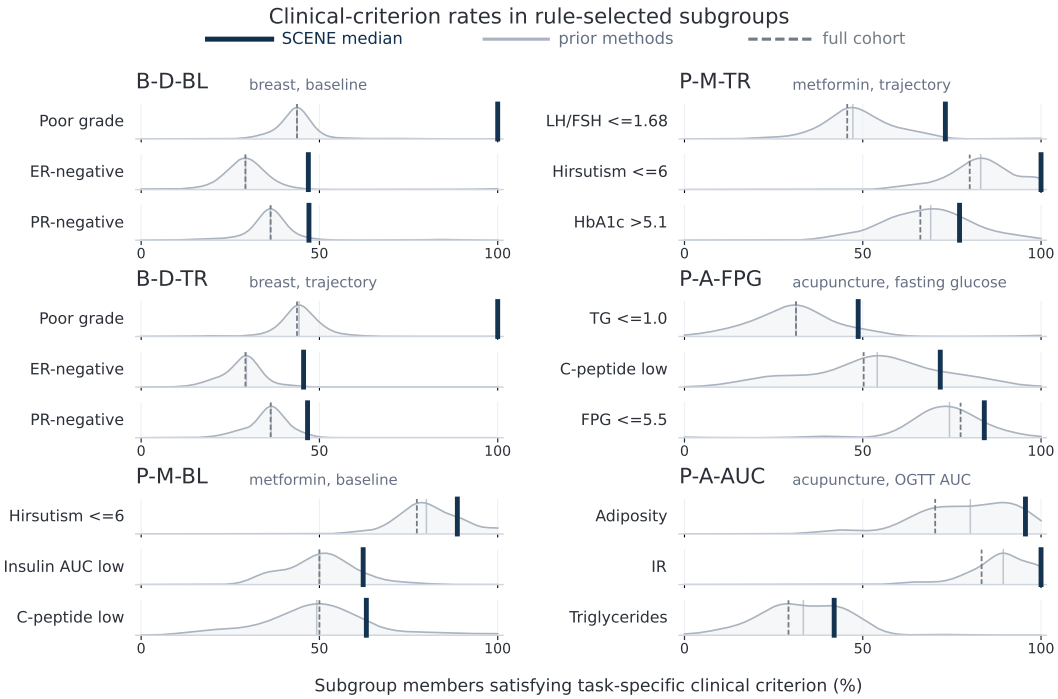


Figure 11: Task-specific clinical-context audit of reported top rules. Each task block contains three endpoint-relevant non-outcome clinical criteria. Dark vertical markers show the median across actual SCENE Best-1 rules outputs of 50 runs; gray densities summarize replayable Best-1 rules of 50 runs from prior methods; the dashed line is the full-cohort criterion rate. The horizontal axis is the percentage of subgroup members satisfying the displayed criterion.

Figure 11 compares actual SCENE Best-1 rule outputs with replayable Best-1 rules from prior methods for each task. For each task-specific clinical criterion, the gray density summarizes the distribution over prior-method top-rule occurrences, while the dark vertical marker reports the median over SCENE top-rule occurrences. We visualize SCENE by its median rather than by a kernel density because the number of SCENE reported top-rule records is smaller than the overall prior-method rule set. The dashed vertical line is the corresponding full-cohort rate. A rightward SCENE median therefore means that SCENE's reported top rules more consistently identify subgroups concentrated on that clinical axis.

Table 17 expands the task-level summary into a full axis-level audit over the predefined non-outcome axes used in Figure 11. This larger table is included for transparency rather than for additional model selection: it shows the cohort rate, the prior-method rule-vocabulary median, and the SCENE median for every predefined clinical axis used in Figure 11. Reporting the complete matrix makes the contextualization claim less dependent on a single aggregate read-out.

Across all six clinical frames, the SCENE median is shifted toward clinically interpretable task-specific phenotypes relative to the prior-method rule vocabulary. The expanded matrix makes the consistency of this shift visible: all 18 predefined axes have positive SCENE–prior margins. The smallest margin is still positive (HbA1c in the metformin trajectory frame, $+8.1$ percentage points), while the largest margins occur on the breast-cancer histologic-grade axis, where SCENE subgroups concentrate poor-grade tumors much more strongly than the replayed prior-method rule vocabulary. The PCOS frames show the same pattern on different clinical axes: metformin tasks shift toward endocrine and insulin-exposure phenotypes, whereas acupuncture glucose and OGTT tasks shift toward glycemic, insulin-resistance, adiposity, and triglyceride axes. Thus the table adds information beyond the task-level summary: it shows that the contextualization advantage is not carried by one favorable axis or one disease frame.

Table 17: Axis-level clinical contextualization matrix. Values are percentages of subgroup members satisfying the displayed non-outcome clinical criterion. "Prior" is the median over replayable top-ranked prior-method rules, and "SCENE" is the median over SCENE top-rule outputs. The axes are fixed contextualization probes and are not used to select or revise rules. This is a post-hoc contextualization audit, not causal feature attribution.

Frame	Clinical axis	Cohort	Prior	SCENE	SCENE–prior	SCENE–cohort
B-D-BL	ER-negative	29.2	29.3	46.9	+17.6	+17.6
B-D-BL	PR-negative	36.3	36.2	47.1	+10.8	+10.7
B-D-BL	Poor grade	43.7	43.7	100.0	+56.3	+56.3
B-D-TR	ER-negative	29.2	29.5	45.6	+16.0	+16.3
B-D-TR	PR-negative	36.3	36.4	46.7	+10.2	+10.4
B-D-TR	Poor grade	43.7	44.3	100.0	+55.7	+56.3
P-M-BL	C-peptide low	50.0	49.2	63.2	+13.9	+13.2
P-M-BL	Hirsutism ≤ 6	77.3	80.0	88.7	+8.7	+11.4
P-M-BL	Insulin AUC low	50.0	50.0	62.3	+12.3	+12.3
P-M-TR	HbA1c > 5.1	66.2	69.0	77.1	+8.1	+11.0
P-M-TR	Hirsutism ≤ 6	80.0	83.1	100.0	+16.9	+20.0
P-M-TR	LH/FSH ≤ 1.68	45.6	47.2	73.2	+25.9	+27.5
P-A-FPG	C-peptide low	50.3	54.1	71.7	+17.7	+21.5
P-A-FPG	FPG ≤ 5.5	77.4	74.3	84.1	+9.8	+6.7
P-A-FPG	TG ≤ 1.0	31.3	31.2	48.7	+17.5	+17.4
P-A-AUC	Adiposity	70.3	80.2	95.6	+15.5	+25.3
P-A-AUC	HOMA-IR ≥ 2.5	83.3	89.4	100.0	+10.6	+16.7
P-A-AUC	TG ≥ 1.7	29.2	33.3	42.0	+8.6	+12.8

This does not establish causal feature attribution or clinical treatment guidance. It supports the more limited claim that SCENE's reported rule vocabulary is more contextualized: the rules that obtain held-out benefit also map to recognizable clinical axes in the corresponding task frame, and this mapping remains visible when the audit is expanded from task averages to the individual clinical probes.

B.3 L1000 Proposition Inspectability

SCENE is designed to return executable propositions rather than only a scalar score or an uninterpreted high-performing subset. In the L1000 setting, this means that a discovered rule should be replayable on held-out perturbational signatures and should identify a concrete region of the assay space that can be inspected by standard LINCS/L1000 quality and response measurements. This appendix evaluates that property from two complementary views: a proposition-level selected-versus-background audit and a target-level replay of top-ranked rule sets. Figure 12 reports the selected-versus-background audit, while Figure 13 visualizes the replayed rule unions in L1000 assay space.

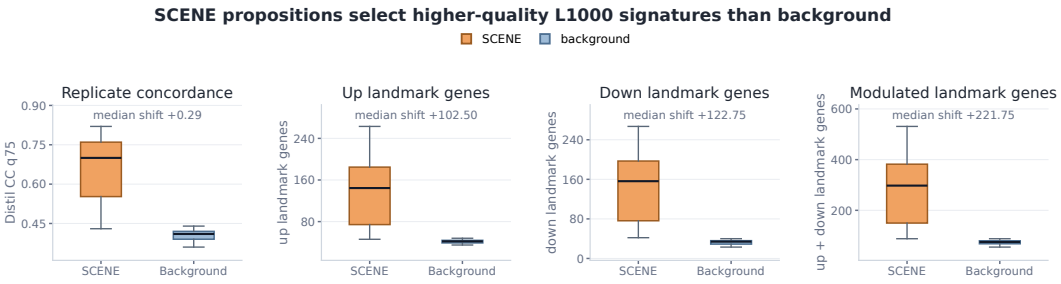


Figure 12: Matched L1000 replay audit. SCENE-selected signatures are compared with each proposition's held-out background using complementary L1000 audit metrics. Panels report replicate concordance, up-regulated landmark genes, down-regulated landmark genes, and total modulated landmark genes.

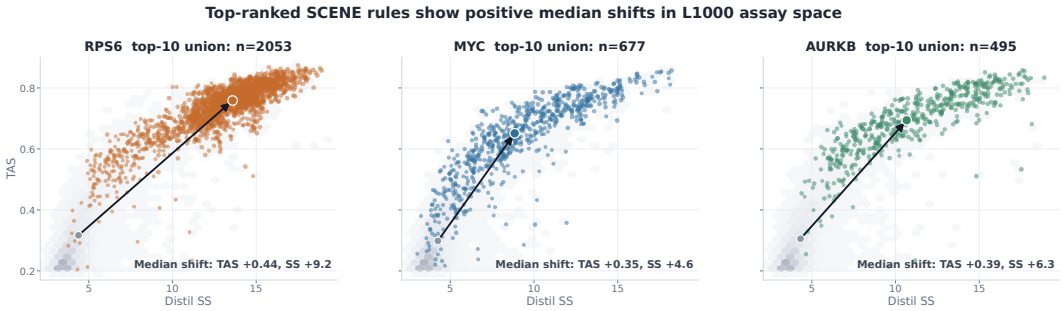


Figure 13: Union of Best-1 full-profile SCENE rules across ten target-seed runs. Colored points are signatures covered by at least one selected SCENE rule; gray hexagons show held-out density and arrows show median displacement.

The selected-versus-background audit in Figure 12 tests whether exported propositions remain meaningful after held-out replay. The unit of the boxplot is a replayed proposition: for each rule, we compute the median metric value among selected signatures and the median value among its corresponding background signatures. The panels use audit metrics that do not duplicate the TAS–Distil SS replay axes: replicate concordance, up-regulated landmark genes, down-regulated landmark genes, and total modulated landmark genes. Across the propositions pooled over RPS6, MYC, and AURKB, median selected-background differences are positive in all four views: $+0.29$ Distil CC q75, $+102.5$ up-regulated landmark genes, $+122.8$ down-regulated landmark genes, and $+221.8$ total modulated landmark genes. The background boxes are narrow because each background is a large, overlapping held-out complement; this stability makes the selected-background separation more interpretable rather than less informative.

Figure 13 complements the proposition-level audit by showing where the selected signatures lie in the assay space. For each target, we take the Best-1 full-profile rule from each of the ten Table 2 target-seed runs and visualize the union of held-out signatures covered by at least one selected rule. The top-rule unions cover 2053 RPS6 signatures, 677 MYC signatures, and 495 AURKB signatures. They shift the covered-signature median by $+0.44$ TAS and $+9.2$ Distil SS for RPS6, $+0.35$ TAS and $+4.6$ Distil SS for MYC, and $+0.39$ TAS and $+6.3$ Distil SS for AURKB. Visually, these median-displacement arrows move toward the high-activity, high-strength region of the L1000 assay space. These positive shifts support the practical inspectability claim: SCENE outputs rules that can be replayed, audited, and interpreted as concrete perturbational contexts.

B.4 Holdout Reliability and Support–Gap Diagnostics

A central failure mode of subgroup discovery is rule overfitting: a rule may have a strong discovery-side contrast but fail when replayed on held-out patients. This section therefore audits rule-level replay records instead of only reporting selected task-level summaries. This audit is especially important for SCENE because the lower-level search explores many candidate rules; a credible discovery system must expose where the archive is stable and where it is support-limited.

Figure 14 shows the held-out effect direction is preserved in 84.9% of evaluations. Direction preservation is 86.4% for full longitudinal contexts and 83.4% for static-only contexts. The median absolute train–holdout gap is 0.071, and the mean absolute gap is 0.175. These values indicate that direction is more reliable than magnitude: many rules preserve whether treatment appears beneficial, but the exact effect size is often attenuated on held-out patients.

Table 18 complements the binned figure from a different angle: it treats held-out smaller-arm support as a reporting boundary and asks how the replay audit changes under increasingly conservative minimum-support floors. The rows are cumulative thresholds. This table is therefore a sensitivity check on how much of the replay archive remains interpretable under stricter support requirements, whereas Figure 14 shows where the support–gap relationship is located. Unlike Table 1 Best-1 DC, which is computed only over committed Best-1 rules, this audit includes the broader eligible replay archive, namely the Top-5 rules from each run.

Table 18: Support-floor sensitivity of held-out replay reliability. The table reports cumulative minimum smaller-arm support floors. Retained archive is shown as a percentage of eligible replay records. The gap is $|\Delta_{\text{train}} - \Delta_{\text{holdout}}|$ and is shown in percentage points.

Support floor	Retained (%)	Direction preserved (%)	Held-out positive (%)	Gap median/P75 (pp)
All replayed	100.0	84.9	91.3	7.1 / 17.5
$m_{\min} \geq 10$	67.4	91.8	93.8	5.0 / 8.4
$m_{\min} \geq 20$	47.9	92.0	94.2	3.9 / 6.9
$m_{\min} \geq 50$	21.5	98.4	100.0	2.0 / 3.2

The sensitivity pattern is useful because it separates two claims that can otherwise be conflated. Directional replay is already high when all eligible rules are considered, but magnitude agreement is support-sensitive. As the support floor becomes stricter, a smaller part of the archive remains, while direction preservation and positive held-out replay increase and the train–holdout gap decreases. This supports a conservative reporting practice: SCENE's replay archive should be interpreted with support-aware uncertainty, rather than treating every discovered rule as equally precise.

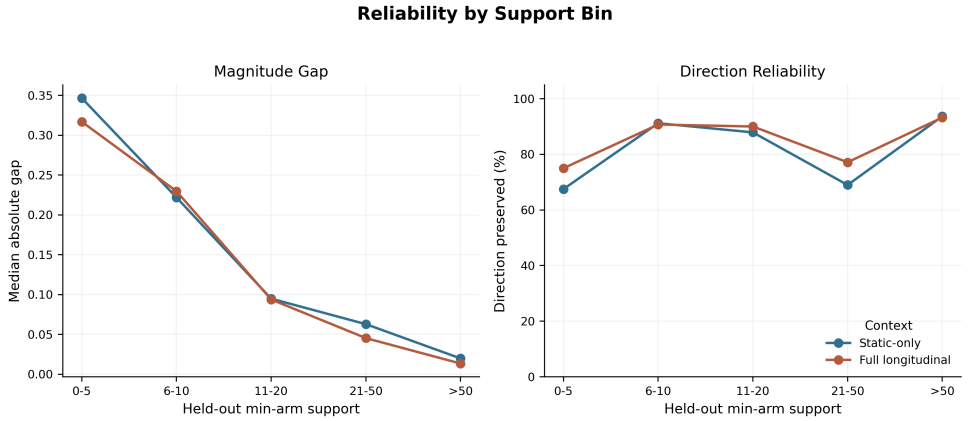


Figure 14: Reliability by held-out support bin. The left panel shows the median absolute train–holdout gap by smaller-arm held-out support. The right panel shows the direction-preservation rate. Magnitude reliability improves sharply with support, while direction reliability is generally higher for supported rules but not perfectly monotonic across intermediate bins.

B.5 Semantic Rule Composition and Feature Provenance

SCENE is designed to produce scenario-grounded rules rather than opaque prediction scores. This section asks what kinds of clinical evidence the discovered rules actually use. We parse clinical rule text into semantic feature families and canonicalize rule predicates to reduce duplicate syntax. This analysis is a syntax and provenance audit, not a causal feature-importance analysis. A rule may contribute to multiple feature families when it combines predicates from multiple evidence sources.

Figure 15 summarizes the SCENE selected rules used in Table 1 as percentages over rule-family assignment flow units. The middle column now shows rule-composition breadth: 3.5% of the displayed assignments come from single-family rules, 46.9% from two-family rules, and 49.6% from rules spanning three or more feature families. On the feature-family side, laboratory measurements are the largest component (37.3%), followed by baseline clinical history (22.7%), generated bins/formulas (20.0%), tumor/pathology descriptors (10.7%), other or unmapped features (7.6%), and longitudinal dynamics (1.6%). Dynamic predicates are therefore present but sparse; the appropriate interpretation is that longitudinal evidence acts as an additional context layer rather than dominating the rule vocabulary. Because multi-family rules contribute once to each used family, the displayed widths are assignment-weighted rather than unique-rule counts. Other includes unmapped or rare feature families.

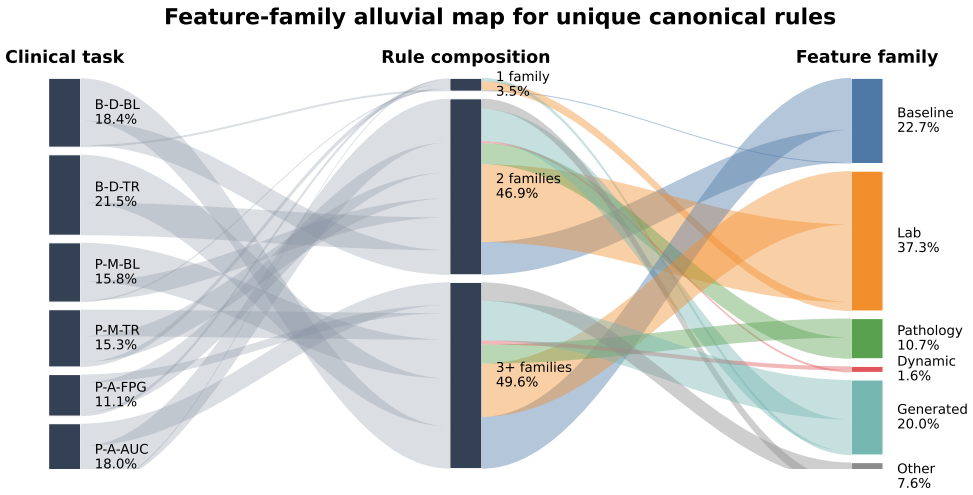


Figure 15: Feature-family assignments from clinical rules. The alluvial map links the six clinical tasks, rule-composition breadth, and feature family. Percentages are computed over rule-family assignments derived from the SCENE selected rules used in Table 1; multi-family rules contribute once to each family they use, and rules appearing in multiple tasks contribute once to each observed task.

Table 19 adds a rule-internal evidence audit that is not visible in the alluvial map. The alluvial figure answers which feature families appear in the archive; the table asks whether recurring semantic archetypes with held-out replay evidence also retain positive clinical effect estimates. We report only aggregate read-outs for archetypes, not raw archive counts. The rows should be interpreted as syntax/provenance summaries of executable rules, not as causal attribution to individual feature families.

Table 19: Held-out evidence among recurring semantic archetypes. Rows summarize canonical rules in the displayed family combination for which held-out effects are available. Benefit is the bad-outcome reduction in percentage points; positive held-out is the share of those rules with benefit greater than zero.

Semantic archetype	Median benefit	Positive held-out	Interpretation
Baseline + Lab	+19.0 pp	94.7%	Patient baseline context combined with direct clinical measurements.
Lab + Generated	+16.9 pp	94.1%	Measured variables augmented by derived scenario-specific summaries.
Baseline + Lab + Generated	+12.6 pp	100.0%	Three-source contextualization linking patient state, measurement, and generated summaries.
Lab only reference	+14.3 pp	72.7%	Single-family laboratory thresholds, included as a shortcut reference.

B.6 L1000 Ablation and Holdout Evidence Diagnostics

The L1000 scenario evaluates whether the same SCENE framework can produce context-bounded target-response findings outside the clinical-trial setting. Here, the relevant outcome is not clinical efficacy but holdout connectivity evidence, target retrieval, support, significance, and generalization. The following diagnostics expand the main ablation table using the same paired L1000 outputs. They are descriptive because they summarize completed, frozen ablation artifacts and do not change the profile selection or replay protocol.

Figure 16 separates the two most important robustness axes: holdout connectivity effect and generalization gap. The favorable region is high DeltaConn and low gap. SCENE (ours) lies closest to this

region in the current run summaries, whereas several ablations retain positive DeltaConn but incur larger generalization gaps.

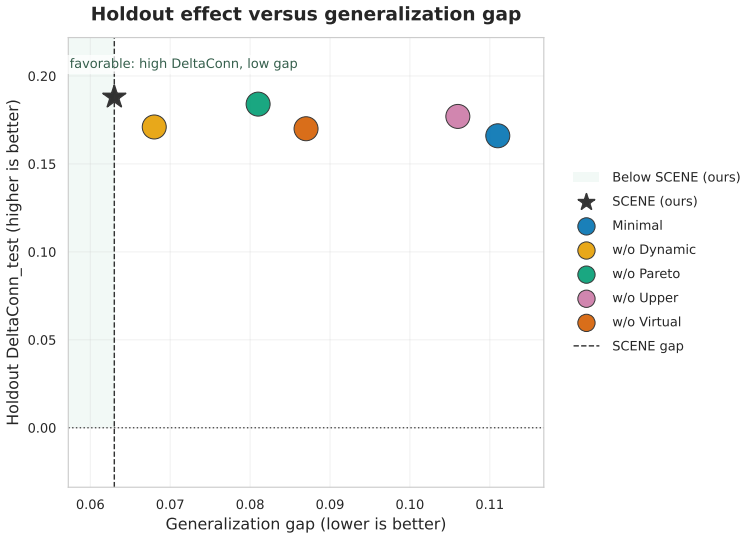


Figure 16: L1000 holdout effect versus generalization gap. Each point is a run-aggregated ablation variant. Higher ΔConn indicates stronger holdout connectivity evidence, whereas lower gap indicates better train–holdout stability. The shaded upper-left region marks the favorable regime. SCENE (ours) combines high holdout effect with the smallest generalization gap among the displayed variants. The dashed vertical line marks the SCENE (ours) gap.

Table 20 gives a minimal-baseline-centered view of the same L1000 ablation evidence. Instead of asking whether each profile has one favorable metric in isolation, it asks which profiles move the system beyond the minimal discovery regime along three core axes: held-out connectivity effect, the strong-positive response tail, and train–holdout stability.

Table 20: Minimal-baseline-centered L1000 effect–stability gains. Relative changes use the minimal baseline as the reference. Higher values are better in all three columns; for the gap column, a positive value means that the train–holdout gap is reduced. The table is descriptive and does not introduce a composite score.

Profile	Effect lift ↑	Strong-tail lift ↑	Gap reduction ↑	Read-out
Minimal	0.0%	0.0%	0.0%	Reference discovery regime.
w/o dynamic	+3.0%	+3.3%	+38.7%	Small effect and tail gains; gap decreases.
w/o virtual	+2.4%	-12.3%	+21.6%	Small effect lift with weaker strong-response tail.
w/o Pareto	+10.8%	-31.9%	+27.0%	Effect lift without strong-tail concentration.
w/o upper	+6.6%	+27.2%	+4.5%	Below SCENE (ours); limited gap reduction.
SCENE (ours)	**+13.3%**	**+50.4%**	**+43.2%**	Largest balanced gains beyond minimal.

Table 20 clarifies the contribution of upper-level planning in the L1000 setting. Without the upper layer, the system remains below SCENE (ours) and provides only a small train–holdout gap reduction, suggesting that lower-level search alone is less stable across the holdout boundary. SCENE (ours) exhibits a different profile: it achieves the largest effect lift, the strongest high-response tail, and the largest reduction in the train–holdout gap relative to the minimal baseline. These coupled gains indicate that upper-level planning helps coordinate contextual proposal generation, search, and validation so that the exported propositions preserve target-response signal across the holdout boundary.

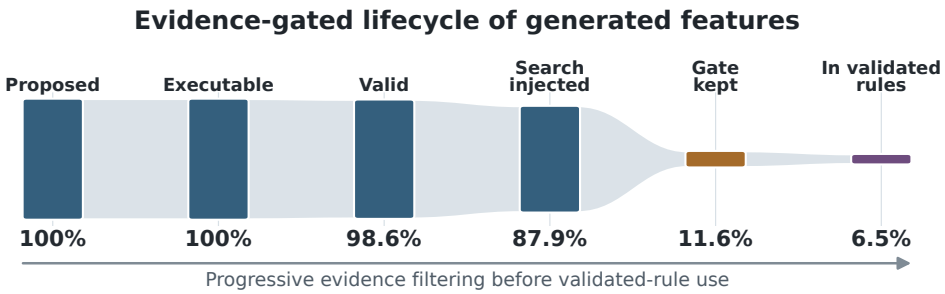


Figure 17: Evidence-gated lifecycle of SCENE-generated features. Percentages are measured relative to all proposed generated run-features. Generated features are first required to be executable and non-degenerate, then injected into search, filtered by an evidence gate, and finally used only if they appear in validated rules. The large drop after search injection indicates that SCENE treats generated features as candidate contextualizations rather than automatically reportable discoveries.

B.7 Evidence-Gated Lifecycle of Generated Features

SCENE allows the discovery process to introduce scenario-specific generated features, such as longitudinal summaries or perturbational signature summaries. This flexibility is useful for knowledge contextualization, but it also requires a strict execution boundary: generated features should not be treated as free-form model suggestions. They must be materialized from registered scenario tables, pass deterministic validity checks, enter the search only as executable feature objects, and survive evidence-based filtering before they can appear in reported rules.

Figure 17 audits this lifecycle over the completed discovery records. The denominator is the set of proposed generated run-features. Nearly all proposals are executable and non-degenerate after deterministic construction, and most are injected into the search frontier. The sharp reduction occurs at the evidence gate: only a small subset is retained after the search evaluates whether the generated feature provides useful frontier evidence. An even smaller subset appears in the final validated rules. This pattern is desirable for SCENE. It shows that the framework uses generated features to expand the grounding space, but does not allow this expanded space to directly determine reported propositions. Instead, generated features must first become valid scenario objects and then earn their place through the same evidence-gated search process as ordinary literals.

This audit supports the main claim that SCENE performs controlled knowledge contextualization rather than unconstrained feature invention. Broad biomedical cues can motivate new scenario-specific summaries, but the executable boundary is governed by the schema, construction grammar, and evidence gate. The final reported rules therefore reflect a conservative subset of generated features: features that are materialized, validated, searched, and retained with supporting evidence. The figure should be read as a protocol audit rather than an additional model-selection step; all quantities are computed from completed runs and are not used to revise the discovery process.

B.8 Grounding Trace from Prior Knowledge to Executable Propositions

The main quantitative results evaluate whether SCENE discovers useful rules or findings, but they do not by themselves show how a broad biomedical direction becomes a concrete scenario-level proposition. Figure 18 illustrates this conversion through four selected examples. Each row connects a semantic direction, grounded feature families, a readable rule description, and the associated validation record. Trajectory examples use predeclared landmarked or trajectory-enhanced features and should be read as predictive subgroup hypotheses, not baseline-only treatment-decision rules.

The four examples were selected to be illustrative but evidence-bearing rather than purely anecdotal. The first three examples come from clinical-trial subgroup discovery and show breast-cancer and PCOS examples in which baseline descriptors, laboratory measurements, longitudinal summaries, generated feature bins, pathology descriptors, or endocrine markers are combined into executable subgroup rules. The last example comes from the L1000 perturbational setting and shows the same grounding idea in a different schema: a target-connectivity direction is instantiated through cell

Knowledge direction	Feature families	Rule summary	Held-out evidence
C1 Breast cancer / longitudinal subgroup			
Early response with endocrine status and baseline burden	Baseline, Generated, Dynamic, Pathology	Low early platelet change Estrogen-receptor positivity absent No injury / poisoning history	RD benefit: **+6.0 pp** Min-arm n: 27 Gap: 0.7 pp
C2 Breast cancer / laboratory-pathology subgroup			
Renal / hepatic context for pathology-defined benefit strata	Lab, Dynamic, Pathology	Higher peak creatinine (cycles 1-4) Poor / undifferentiated tumor grade Alkaline phosphatase not elevated	RD benefit: **+13.3 pp** Support n: 15 Gap: 1.6 pp
C3 PCOS / metformin-response subgroup			
Endocrine phenotype for metformin response	Baseline, Generated, Lab	Low baseline LH bin Lower baseline hirsutism score Total testosterone observed	RD benefit: **+12.6 pp** Support n: 15 Gap: 18.7 pp
C4 L1000 / perturbation subgroup			
Pathway state and signature strength for target connectivity	Cell context, Pathway program, Signature magnitude	Non-CD34 cell context High dose-range signature span Reduced DNA-repair-like program	Connectivity Δ: **+0.249** Subgroup n: 292 Gap: 0.029

Figure 18: Illustrative grounding paths. Each row links a concise biomedical direction, feature families, a rule summary (all conditions apply), and archived validation evidence. Clinical risk-difference (RD) benefit is in percentage points (pp); L1000 reports a connectivity delta. Gap is the absolute training/discovery–validation effect difference. All four cases preserve effect direction. These are selected examples, not an additional model-selection procedure.

context, pathway-program state, and transcriptional signature magnitude. In all four cases, the output is not just a natural-language explanation. Each displayed rule description corresponds to an archived predicate over scenario-visible features, and the rightmost column reports post-discovery evidence on held-out or validation data.

This organization clarifies SCENE's knowledge-contextualization process. A static knowledge-injection approach can state that a biomedical concept is relevant, and a data-only rule miner can return a high-scoring rule, but neither necessarily records how the concept is grounded into the current schema. The figure illustrates the division of work in SCENE: a direction proposes the semantic intent, grounding maps it to feature families, and search turns those families into a concrete rule. The clinical examples report positive archived validation estimates of risk-difference benefit, while the L1000 example shows a positive connectivity delta for a context-specific perturbational subgroup. These examples therefore make the reported propositions inspectable.

The figure is not intended as an aggregate performance comparison and should not be interpreted as independent biomedical validation of the displayed mechanisms. Rather, it is a qualitative audit of the reporting boundary. It demonstrates that SCENE exports propositions with a visible grounding path, rather than reporting isolated rule strings or post-hoc textual rationales.

Fixed-input planner calls. On P-M-TR, seed 58, we compare the earliest three successful API calls in chronological order, with identical source data, split, LLM, temperature zero, and no fallback. All calls retain an insulin-resistance component, while the accompanying direction differs for Call 1 (Table 21). Calls 2 and 3 produce identical structured groundings. Across the three call pairs, grounded-feature Jaccard averages 0.810 and ranges from 0.714 to 1.000. This small check distinguishes shared semantic focus from exact output determinism. Conditioning on successful calls does not measure API failure rates. The 0.097 split-level ARR standard deviation in Figure 9 describes the entire discovery pipeline across patient splits, not planner variance alone.

Exploratory report quality. We freeze one report for each method and clinical frame for SCENE and the baselines across six clinical frames (48 reports per evaluator). Two AI evaluators and one human expert rate all reports on the four dimensions in Table 22. Each cell averages six reports. Best non-SCENE selects the highest baseline mean separately for each evaluator and dimension.

Table 21: Planner consistency.

Call	Grounding: Direction	Grounding: Feat.	J
1	MR + IR	12	0.714
2	HA + IR	12	0.857
3	HA + IR	12	0.857

Table 22: Report-quality ratings.

Rater	SCENE C	SCENE E	SCENE A	SCENE F	Best non-SCENE C	Best non-SCENE E	Best non-SCENE A	Best non-SCENE F
AI-1	4.33	4.67	5.00	4.83	3.83	2.83	3.83	3.67
AI-2	4.33	4.83	5.00	4.83	3.67	2.83	4.00	3.33
Human	4.50	5.00	5.00	5.00	4.00	3.33	4.17	3.67

MR, HA, and IR denote metabolic reserve, hyperandrogenism, and insulin-resistance magnitude. Feat. counts grounded features, and J is the mean feature-set Jaccard with the other two calls. Rating dimensions are clinical/biological coherence (C), evidence sufficiency and stability (E), auditability and replayability (A), and readiness (F).

SCENE receives higher mean ratings on all four dimensions from each evaluator. This is exploratory evidence about the presentation and inspectability of the reports. A single human evaluator, without uncertainty estimates or a prospective workflow comparison, cannot establish general expert preference, biological validity, or readiness for clinical use. The audit does not compare SCENE with hypotheses generated by an expert-led discovery workflow.

B.9 Knowledge-Guided Controls and Prior Sensitivity

We test whether initial knowledge guidance or a single proposal pass can account for SCENE's gains, and whether explicit prior records contribute to them. The controls below complement Tables 1 and 2 with full-precision results and paired uncertainty. The preceding replay and component analyses retain their original comparison sets.

Paired uncertainty. The intervals below are 95% percentile-bootstrap intervals obtained by resampling matched run units and recomputing the mean between-method difference. For the two MaxTE contrasts in Table 23, we pair identical patient split identifiers within each frame (50 splits). The L1000 comparison pairs episodes by target and seed (three targets, ten seeds each). These intervals describe variation under the fixed benchmark protocol. Repeated clinical splits share patients, and L1000 seeds reuse the same three targets.

Initial grounding for clinical rule search. As a targeted control for static initial knowledge guidance, *MaxTE + Grounding* receives only the raw-feature set in SCENE's validated generation-0 grounding plan. Relative to unrestricted MaxTE, it keeps the patient split, endpoint, rule-length limit, MaxTE settings, and held-out evaluator fixed. It receives no SCENE thresholds, generated features, later-round feedback, or held-out outcomes. Table 23 gives the full reported precision and both paired contrasts. Grounding alone improves MaxTE's point estimate on three frames and lowers it on three. SCENE has the highest point estimate on all six. The intervals for SCENE minus Grounded MaxTE exclude zero on the four PCOS frames and overlap zero on the two breast-cancer frames. This compares static grounding with the full pipeline.

Table 23: Clinical control means and paired 95% intervals for held-out bad-outcome ARR. S, M, and G denote SCENE, unrestricted MaxTE, and MaxTE + Grounding. Differences are in ARR units.

Frame	SCENE	MaxTE	Grounded	$S - M$: 95% CI	$S - G$: 95% CI
B-D-BL	0.0860	0.0760	0.0015	$[-0.0385, 0.0985]$	$[-0.0193, 0.1357]$
B-D-TR	0.0950	0.0120	0.0265	$[0.0008, 0.2008]$	$[-0.0079, 0.1806]$
P-M-BL	0.4410	0.2340	0.1704	$[-0.0083, 0.3210]$	$[0.1532, 0.3201]$
P-M-TR	0.4390	0.1210	0.1721	$[0.1360, 0.4621]$	$[0.1315, 0.3130]$
P-A-FPG	0.2690	0.1170	0.1210	$[0.0783, 0.2347]$	$[0.0648, 0.1994]$
P-A-AUC	0.2700	0.1320	0.1009	$[0.0678, 0.2543]$	$[0.0933, 0.2040]$

Grounded MaxTE has P-ARR 0.03506, U-Supp. 50.67%, and Best-1 direction consistency 66.33%, versus 0.093, 84.7%, and 100.0% for SCENE. These diagnostics follow the metric definitions and

Best-1 selection and held-out replay protocol used in Table 1. Paired intervals are provided for the two MaxTE contrasts, not for every original baseline.

Source-documented prior records. We remove only the explicit source-documented prior records while keeping the schema, task context, LLM, search budget, split, and evaluator fixed. Table 24 reports lower ARR on the two tested frames, with positive paired intervals for Full minus No Sources. This supports a contribution from the supplied records in these frames. The control retains the LLM's parametric knowledge and does not measure sensitivity to all possible priors or across the full benchmark.

Table 24: Targeted removal of source-documented prior records. No Sources removes the explicit records only. Δ is Full minus No Sources in held-out bad-outcome ARR.

Frame	Full	No Sources	Δ	95% paired CI
B-D-BL	0.0860	0.0787	0.0073	[0.0008, 0.0217]
P-M-TR	0.4390	0.3637	0.0753	[0.0432, 0.1521]

One-shot L1000 proposals. The One-shot LLM Proposal control uses the same source priors, schema, initial candidate budget, split, output type, and Top-K evaluator as SCENE, on RPS6, MYC, and AURKB with ten paired seeds per target. It performs one proposal/grounding pass, validates and ranks the candidates using training-side evidence, then freezes and replays them. It has no later evolution or evidence feedback. The initial proposal budget is matched, but total adaptive computation is not.

Table 25: One-shot L1000 control and paired 95% intervals. The contrast is SCENE minus One-shot except for Gap, where it is One-shot minus SCENE. Positive differences favor SCENE. Intervals for percentage metrics are in percentage points (pp).

Metric	SCENE	One-shot	95% CI
Conn. ↑	0.188	0.107	[0.0173, 0.1364]
Strong-Pos. ↑	41.5%	24.42%	[3.20, 25.01] pp
AUPRC ↑	0.268	0.2532	[−0.0209, 0.0181]
U-Supp. ↑	31.4%	38.31%	[−10.1211, 8.3101] pp
Gap ↓	0.063	0.129	[0.0042, 0.2179]

Table 25 supports gains in connectivity, strong-response concentration, and stability from training to holdout under this protocol. The AUPRC and U-Supp. intervals include zero. The one-shot control has the higher U-Supp. point estimate. These results do not establish superiority on every metric or replace a comparison with specialized perturbational retrieval methods.

C LIMITATIONS

SCENE is designed for biomedical hypothesis generation. The clinical analyses identify scenario-grounded subgroups with held-out outcome contrasts, but these propositions are not diagnostic criteria, treatment recommendations, or patient-level decision rules without independent validation. Live language-model calls can propose different directions or grounding cues across providers, model versions, and decoding settings. Similarly, L1000 replay provides perturbational assay evidence for prioritizing follow-up, not wet-lab mechanism confirmation or drug-efficacy evidence. The experiments cover the clinical task frames and L1000 target programs studied in this paper; applications to new endpoints, cohorts, assays, or disease areas should be accompanied by task-specific validation and domain review. These results should therefore be interpreted as evidence that broad biomedical knowledge can be contextualized into auditable propositions, rather than as a substitute for prospective clinical or experimental studies.

D BROADER IMPACTS

SCENE may improve biomedical discovery workflows by making hypothesis generation more transparent: outputs are expressed as executable rules with source-traceable search directions, support/effect summaries, and replay diagnostics that can be inspected by domain experts. This can help researchers organize large clinical or perturbational datasets and prioritize candidates for follow-up analysis. More broadly, SCENE points toward auditable auto-research workflows in which agents propose executable hypotheses rather than ungrounded claims; however, such automation can also create automation bias if concise rules are trusted without expert review. The main risk is overinterpretation of concise rules as actionable medical or therapeutic claims. Responsible use therefore requires expert review, independent validation before downstream decisions, and appropriate governance for data privacy, security, and fairness when patient data are involved. Clear reporting of validation scope and uncertainty diagnostics is important to keep the resulting propositions in their intended exploratory role. Finally, multi-agent discovery has computational and operational costs. Live LLM calls, evolutionary search, and repeated validation can increase compute use and complicate reproducibility. The implementation choices in this paper emphasize typed role outputs, deterministic validation, and reusable manifests so that future users can limit unnecessary calls, replay completed runs, and audit the provenance of reported propositions.