Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Knowledge-grounded Agentic Framework for Single-cell RNA Sequencing Method Hypothesis Generation

Zhen Chen

z.chen@polyu.edu.hk

The Hong Kong Polytechnic University

Xuewen Zhou

The Hong Kong Polytechnic University

Haiman Guo

The Hong Kong Polytechnic University

Lichao Sun

Lehigh University https://orcid.org/0000-0003-1539-7939

Article

Keywords:

Posted Date: November 19th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11760251/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: There is **NO** Competing Interest.

A Knowledge-grounded Agentic Framework for Single-cell RNA Sequencing Method Hypothesis Generation

Xuewen Zhou[1] **Haiman Guo**[1] **Lichao Sun**[2] **Zhen Chen**[1,*]
[1] The Hong Kong Polytechnic University [2] Lehigh University
{z.chen@polyu.edu.hk}

Single-cell RNA sequencing (scRNA-seq) analysis relies on computational methods that shape biological conclusions. Developing new methods requires identifying computational operations that address problem constraints to produce intended effects. Here we investigate coordinated agents that extract methodological reasoning from full-text literature and reuse it as prior knowledge for method hypothesis generation. We instantiate this approach as scHypothesizer, a knowledge-grounded agentic framework that uses Mechanism-centered Generation (MCG) to formulate hypotheses from literature-derived Research Primitive Knowledge Graphs (RPKGs) and subjects them to early empirical assessment. We show that coordinated agents recover source-supported reasoning and that full-text RPKGs preserve more complete reasoning than abstract-only representations. In time-filtered retrospective evaluation of 2,429 Problems, MCG achieved the best ranking performance among compared retrieval methods, with a query-macro AUROC of 0.783 and Hit@10 of 69.1%. Finally, scHypothesizer supported early empirical assessment of four researcher-approved hypotheses, providing informative evidence regarding intended Effects, the settings in which they hold, and associated limitations. Together, these results suggest that coordinated agents can make methodological reasoning from prior studies reusable as source-traceable knowledge for generating new method hypotheses and informing their early empirical assessment.

1. Introduction

Single-cell RNA sequencing (scRNA-seq) has advanced understanding of biological processes in health and disease by resolving heterogeneous cell populations and states [1, 2]. Improvements in throughput and the growth of large-scale data resources have further expanded the scale and availability of single-cell datasets [3, 4]. The biological conclusions drawn from these data, however, depend strongly on how the data are computationally processed and interpreted [5–7]. Computational methods shape how cells are represented, annotated, and corrected for unwanted variation. These transformations determine which biological signals are preserved, distorted, or lost during downstream analysis. Even capable methods exhibit consequential limitations. Integration can obscure rare or batch-specific populations [5]; reference mapping can produce overconfident assignments when query cells represent types or states absent from the reference [8]; and ambient-RNA correction can remove genuine marker expression [9]. Addressing such failures requires more than proposing a new algorithm. It requires researchers to identify a specific unresolved problem, reason about which mechanism could plausibly address it, and define an empirical test of the expected effect.

Developing a computational method therefore involves three closely linked activities: surveying the methodological literature, formulating a method hypothesis, and testing that hypothesis through implementation and benchmarking. Agents powered by large language models (LLMs) could support each of these activities. They can generate scientific proposals [10, 11], plan and conduct experiments [12, 13], search over programs [14], and participate in multi-agent scientific reasoning [15–18]. Applications of these capabilities have been reported in domains ranging from atomic force microscopy

[19] and computational software [20] to cancer pathology [21]. Yet the ability to produce a convincing proposal does not establish that the proposal addresses a documented problem, that its mechanism is relevant to that problem, or that it performs as claimed. Model-generated research ideas can be infeasible, unsupported, or based on hallucinated information [22, 23]. The central challenge is therefore not simply to generate plausible method ideas, but to connect a problem documented in the literature to a relevant mechanism and then to an empirical assessment capable of supporting or refuting the resulting claim.

We define a computational method hypothesis as a testable proposition that links three elements: a specific limitation of existing methods in a defined task, a computational mechanism intended to address that limitation, and a measurable effect expected from the mechanism. Here, a mechanism refers to an algorithmic, statistical, or data-processing component rather than a biological process. These three elements are inseparable: a problem becomes actionable only when paired with a mechanism that could plausibly alter the relevant outcome, and a proposed mechanism becomes empirically testable only when its intended effect can be evaluated against established methods. Therefore, a method hypothesis specifies in advance what assessment evidence would support or refute its central claim.

The underlying methodological reasoning that drives method development by domain experts is already distributed across the literature. Method papers typically describe the limitation that motivated the work, the mechanism introduced in response, and the effect that the mechanism was intended to produce. However, these relations remain embedded in prose, making them difficult to search, compare, or recombine systematically. Previous studies have shown that scientific literature can be converted into structured computational representations [24–26]. Structured information extraction has produced machine-readable experimental records [24], while KARMA uses specialized agents to identify entities, extract relations, and resolve conflicts [25]. Knowledge graphs have also supported question answering in materials science [27], anticipated later research directions [26], and connected biomedical entities such as genes, diseases, and drugs [28–30]. While these studies demonstrate that scientific knowledge can be made accessible to downstream computation, existing schemas fall short of capturing the core reasoning required for computational method development. This gap motivates our central question: can we formalize methodological reasoning around unresolved problems, candidate mechanisms, expected effects, and empirical evaluation schemes, and thereby guide agents to extract this reasoning from full-text literature and reuse it for method hypothesis generation?

Here we investigate the use of coordinated agents to extract methodological reasoning from full-text literature and reuse it as prior knowledge for method hypothesis generation. We instantiate this approach as scHypothesizer, a knowledge-grounded agentic framework that coordinates these activities through a shared knowledge-graph schema. Starting from a researcher-defined scRNA-seq method-development topic, its agents retrieve relevant literature and construct Research Primitive Knowledge Graphs (RPKGs) from full-text articles. These graphs represent methodological reasoning as source-supported links connecting a methodological Problem to the computational Mechanism used to address it and to the intended or observed Effect of that Mechanism, making this reasoning available for reuse. scHypothesizer then uses Mechanism-centered Generation (MCG) to formulate candidate *method hypotheses* grounded in the RPKGs, each specifying a target Problem, a proposed Mechanism, and an intended Effect. Following researcher approval, scHypothesizer assists with implementing selected hypotheses and conducting early empirical assessments to inform decisions about further development.

To evaluate this approach, we constructed RPKGs for three topics: rare-cell-preserving ambient RNA decontamination; reference mapping with calibrated uncertainty under donor and study shifts;

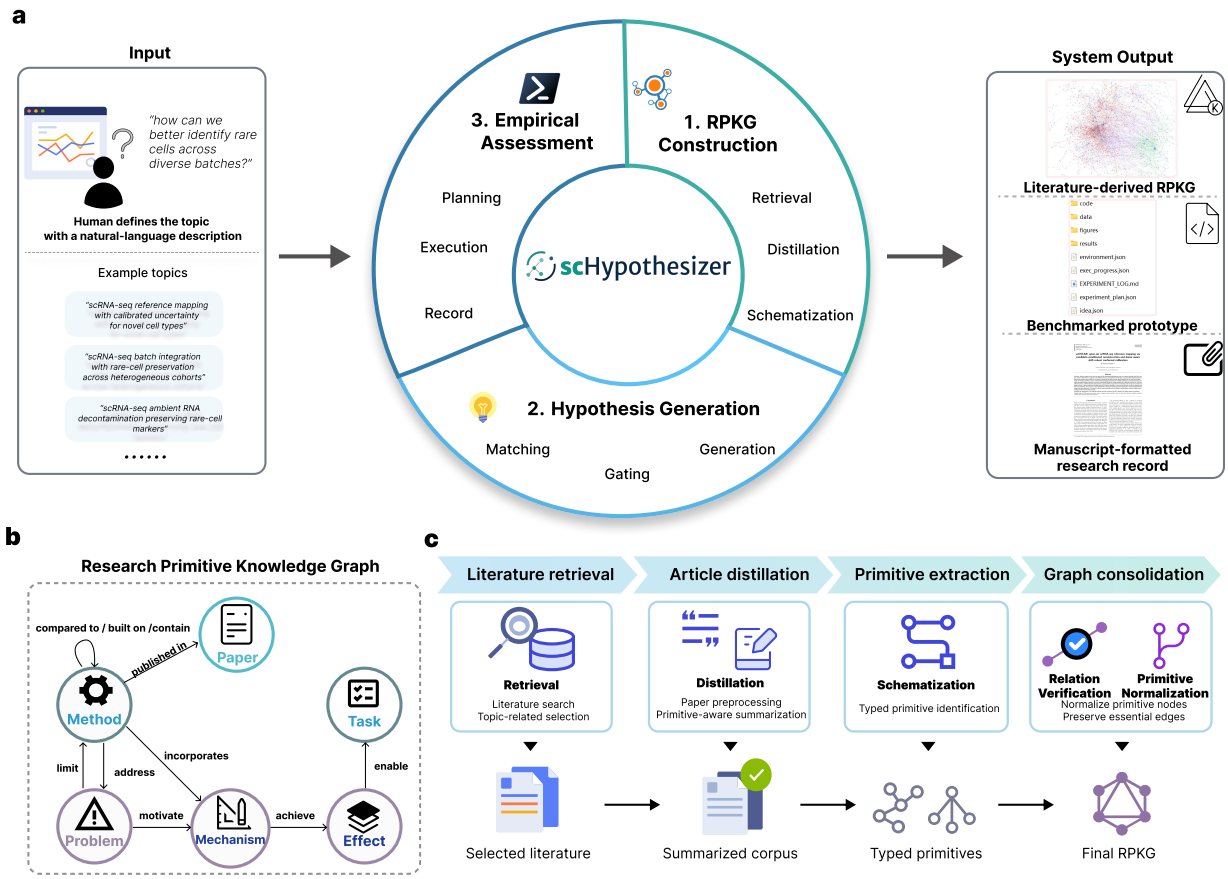


Figure 1 | Overview of scHypothesizer and RPKG Construction. **(a)** A natural-language topic defined by human researchers initiates three stages: RPKG Construction, Hypothesis Generation, and Empirical Assessment. The system returns a literature-derived RPKG and candidate method hypotheses; for researcher-authorized candidates, it additionally returns empirical assessment outputs assembled into a research record. **(b)** RPKG schema linking Paper, Method, Problem, Mechanism, Effect and Task nodes through typed relations. **(c)** RPKG Construction workflow for literature retrieval, article distillation, primitive extraction, relation verification, and primitive normalization.

and rare-cell-preserving batch integration. We also constructed a domain-scale graph covering diverse scRNA-seq methods. We show that coordinated agents reliably construct source-supported RPKGs and preserve Problem–Mechanism–Effect reasoning more completely from full texts than from abstracts. A time-filtered evaluation of 2,429 problems then assessed the retrieval of earlier mechanisms corresponding to solutions reported in later publications. Our approach achieved the highest ranking performance among the compared retrieval methods, supporting the use of historical mechanisms as candidate prior knowledge for method development. Candidate comparisons further examined how these priors shaped hypothesis formulation. Finally, four case studies illustrated how early empirical assessments could provide informative evidence for researchers deciding whether to pursue further development.

2. Results

2.1. Coordinated agents reliably construct source-traceable RPKGs

RPKGs are intended to preserve methodological reasoning from the literature as source-supported knowledge for subsequent hypothesis generation, but errors in RPKGs could lead to hypotheses based on incorrect information, making reliable RPKG construction essential. scHypothesizer uses coordinated agents to construct RPKGs from researcher-defined topics (Fig. 1a). As shown in Fig. 1c, the agents retrieve and screen the literature, summarize full-text articles, extract methodological primitives, verify relations, and normalize primitives. In order to ensure reliability, RPKG construction should meet two requirements: the selected literature should match the intended methodological scope, and the extracted relations should be supported by their source articles. We therefore used human audits to assess whether the screening decisions and extracted relations met these requirements when RPKGs were constructed with general-purpose LLMs without topic-specific model training. The audits covered three tested scRNA-seq topics: rare-cell-preserving ambient RNA decontamination (Ambient), reference mapping with calibrated uncertainty under donor and study shifts (Ref. map), and rare-cell-preserving batch integration (Integration).

RPKG construction requires source articles that contain methodological reasoning relevant to the research topic. We therefore first examined whether the literature selected by the Retrieval Agent matched the intended methodological scope. Across the three topics, human reviewers judged 575 of the 595 screening decisions (96.6%) to be correct. Most errors involved retaining articles outside the intended scope. For example, the agent retained DAGFormer [31] because its mechanism was relevant to integration. However, DAGFormer was designed for single-cell drug-response prediction instead of batch integration. The audit also found that some relevant articles had been excluded, including a spatial reference-mapping study that met the inclusion criteria. Thus, the Retrieval Agent generally selected literature within the intended scope, providing suitable source articles for subsequent RPKG construction.

After confirming that the selected literature generally matched the intended scope, we next examined whether the relations stored in the RPKGs were supported by their source articles. Relations without source support could introduce methodological reasoning that was not reported in the literature. During RPKG construction, the Relation Verification Agent checked the extracted relations and decided which to retain in the final graph. To evaluate relation verification, human reviewers independently checked three samples against their source articles. The samples contained 250 raw candidate relations before verification, 250 retained relations, and 250 discarded relations, giving 750 relations in total. The reviewers found source support for 99.6% of retained relations, compared with 95.2% of raw candidate relations and 90.4% of discarded relations (Fig. 2e). Thus, a higher proportion of relations had source support in the retained sample than in the raw candidate sample. Most discarded relations were also source-supported, which is consistent with the verification score penalizing low topical relevance as well as weak textual support. Nearly all retained relations examined in the audit were supported by their source articles, providing evidence for the reliability of the methodological relations stored in the RPKGs.

The experiments above mainly used GPT-5.4. We therefore examined whether the reliability of RPKG construction depended strongly on the LLM backend. For the same three topics, we repeated extraction with GPT, Opus, and Qwen and asked human reviewers to check the extracted nodes and relations against their source articles. The human audits showed that most nodes and relations were extracted correctly, with pooled node-extraction accuracy of 98.3% and pooled edge-extraction accuracy of 87.6% (Fig. 2f). We further compared the relations extracted by each pair of models to determine whether different backends produced conflicting claims. Although the models did

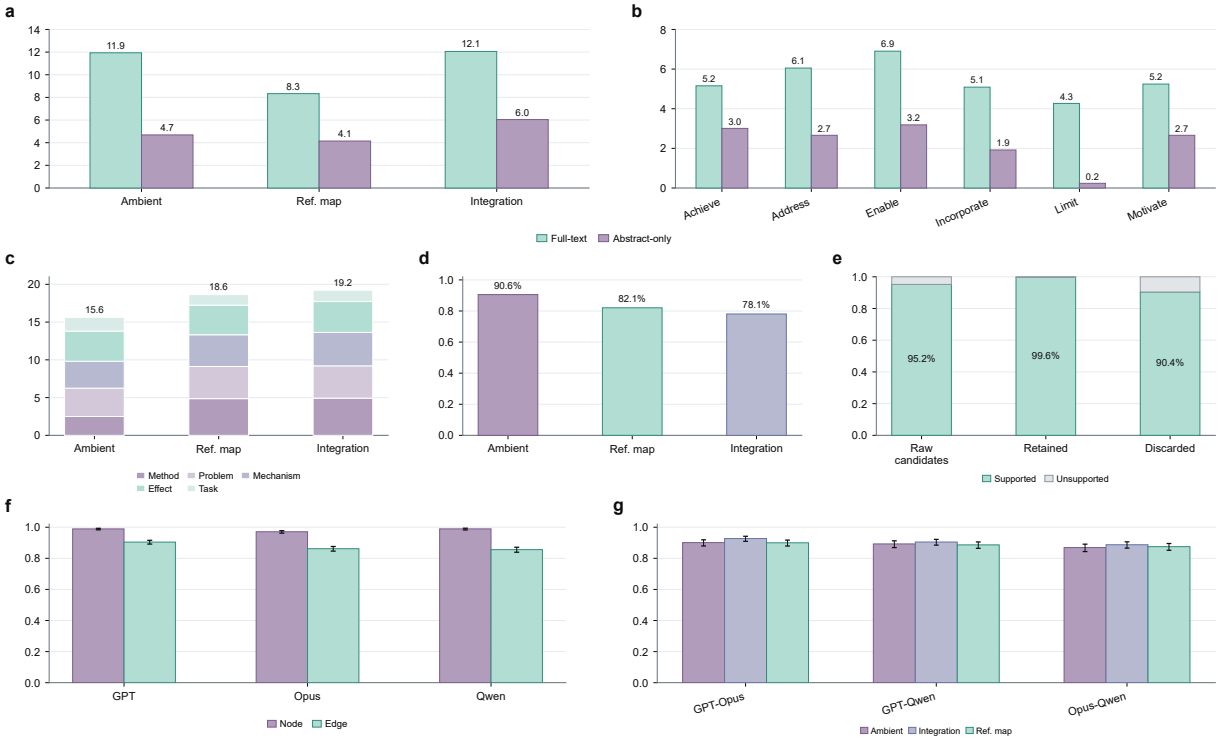


Figure 2 | Full-text RPKG Construction preserves source-supported methodological reasoning. (a) Complete Problem–Mechanism–Effect–Task paths recovered per paper from full-text and abstract-only extraction across 135 matched papers (16 Ambient, 48 Ref. map and 71 Integration). **(b)** Functional relations recovered per paper by relation type in the same matched corpus. **(c)** Connected Method, Problem, Mechanism, Effect and Task nodes retained per paper across 149 eligible full-text articles (16 Ambient, 50 Ref. map and 83 Integration). **(d)** Fraction of retained edges representing functional methodological relations in the same 149-paper corpus. **(e)** Fraction of 250 raw candidate, 250 retained and 250 discarded relations supported by their attributed source articles. **(f)** Node- and edge-extraction accuracy for GPT, Opus and Qwen, pooled across the three topics. **(g)** Pairwise non-conflict edge agreement between language-model backends across the three topics. Error bars in **(f)** and **(g)** indicate Wilson 95% confidence intervals.

not extract identical sets of relations, pairwise non-conflict edge agreement ranged from 87.8% to 91.0% after pooling across topics (Fig. 2g). Thus, most comparisons found no conflicting claims, even when the models differed in which relations they extracted. Overall, these results show that scHypothesizer can use coordinated agents to select relevant literature and construct RPKGs containing methodological relations supported by their source articles. The extraction results across the three topics and three LLM backends further support the reliability of RPKG construction in the tested settings.

2.2. Full-text RPKGs preserve methodological reasoning beyond abstracts

Building upon our reliability analysis, we evaluated the methodological content captured within the RPKGs. RPKGs represent methodological reasoning through functional relations linking Problems, Mechanisms, Effects, and Tasks. However, abstract-only extraction often misses these multi-element connections because abstracts typically omit detailed methodological context. We reasoned that full-text articles provide the necessary information to reconstruct complete reasoning paths while

maintaining logical coherence. We therefore asked two questions: whether full-text extraction recovers more complete Problem–Mechanism–Effect–Task paths than abstract-only extraction across matched papers, and whether the recovered paths form coherent methodological reasoning rather than arbitrary combinations of primitives.

To address the first question, we compared full-text and abstract-only extraction across 135 matched papers (16 for Ambient, 48 for Ref. map, and 71 for Integration). Full-text extraction recovered more than twice as many complete Problem–Mechanism–Effect–Task paths per paper as abstract-only extraction, averaging 10.8 paths compared to 5.0 across all topics (Fig. 2a). This increase was consistently observed across all three individual topics. Functional relations followed a similar trend, averaging 5.5 per paper for full text versus 2.3 for abstracts across the six relation types (Fig. 2b). The advantage of full-text extraction extended to every relation type. In particular, full-text extraction captured markedly more detailed design relations: *limits* relations averaged 4.3 per paper for full text compared with only 0.2 for abstracts, while *incorporates* relations averaged 5.1 and 1.9, respectively.

These additional relations captured nuanced methodological details that were absent from abstracts. For example, in SIDA [32], abstract-only extraction recovered no *limits* relation and only one *incorporates* relation, whereas full-text extraction recovered 14 and 7, respectively. These relations detailed specific limitations of unsupervised comparators as well as mechanism choices involving cross-batch cell pairing and semantic alignment losses. Across all 149 eligible full-text articles, the resulting RPKGs consistently incorporated all five methodological primitive types—Method, Problem, Mechanism, Effect, and Task—with functional relations accounting for 78.1–90.6% of retained edges across the three topics (Fig. 2c,d). Thus, full-text extraction recovered not only a higher volume of relations, but also the structural context connecting motivating problems, algorithmic designs, and intended effects.

To address the second question, we evaluated whether these recovered relational chains formed coherent methodological reasoning rather than arbitrary combinations of individually supported relations. Across the three topics, full-text RPKGs contained 193 (Ambient), 408 (Ref. map), and 989 (Integration) complete Problem–Mechanism–Effect–Task paths. We compared observed paths against paper-stratified shuffled controls, which preserved each paper's primitive inventory while randomly reassigning its internal relations. This negative control held paper-specific concepts constant while selectively disrupting their relational structure.

In a blinded human audit, 126 of 150 observed paths (84.0%) were judged logically coherent, compared with only 46 of 150 shuffled controls (30.7%). For example, an observed path for QClus [33] correctly linked doublet-like artifacts (Problem) to post-clustering doublet filtering (Mechanism) and the intended reduction of those artifacts (Effect). In contrast, the corresponding shuffled control erroneously paired the same Problem with unrelated nuclear-contamination features. The marked difference between observed and shuffled paths indicates that full-text RPKGs not only retain substantially more methodological reasoning than abstract-only representations, but also preserve structurally coherent reasoning paths linking Problems, Mechanisms, Effects, and Tasks. These results indicate that full-text extraction by coordinated agents captures more complete and more coherent methodological reasoning than abstract-only extraction.

2.3. RPKG supports the reuse of methodological reasoning as prior knowledge for later method development

Preserving methodological reasoning in RPKGs enables scHypothesizer to systematically reuse historical Mechanisms as prior knowledge for new method-development problems. Under a retain-then-reuse

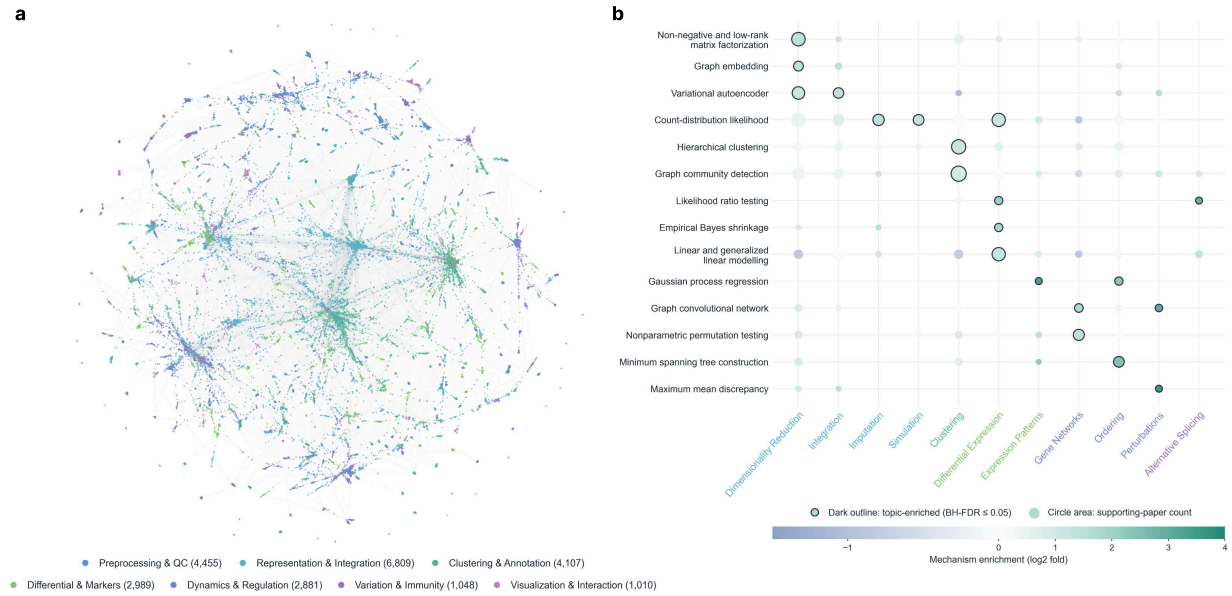


Figure 3 | Topic organization of the domain-scale RPKG. (a) Force-directed layout of connected RPKG nodes, colored by scRNA-tools topic family. **(b)** Enrichment of Mechanism families across scRNA-tools topics. Circle area indicates the number of supporting papers, color indicates $\log_2$ fold enrichment, and dark outlines mark statistically significant associations at Benjamini–Hochberg adjusted $q \leq 0.05$.

paradigm, scHypothesizer employs Mechanism-centered Generation (MCG; Methods) to pair historical Mechanisms with newly defined Problems. To systematically evaluate the feasibility and effectiveness of this approach at scale, we constructed a domain-scale RPKG and implemented a temporal evaluation framework. This retrospective setup restricts candidate Mechanisms to those published before each Problem's publication year, directly testing whether historical Mechanisms offer plausible solutions for future method-development Problems.

Leveraging scHypothesizer's construction pipeline, we built a domain-scale RPKG from 1,093 single-modality scRNA-seq method papers indexed by the scRNA-tools database [34]. The resulting graph contained 23,299 connected nodes, including 6,046 Mechanisms, linked by 47,459 verified functional relations. To observe how these Mechanisms correspond to specific computational tasks, we identified 28 recurrent Mechanism families. Across topics and families, 20 topic–family pairs were significantly enriched (Fig. 3b). For example, dimensionality-reduction methods preferentially incorporated matrix factorization and variational autoencoders, whereas differential-expression methods relied on generalized linear modeling and empirical Bayes shrinkage. These associations are consistent with established usage of these mechanisms, providing a sanity check on the domain-scale graph.

With the domain-scale RPKG constructed, we observed a difference in how Problems and Mechanisms are described across method papers. A Problem cannot be interpreted in isolation; it relies on its graph neighbors to specify what fails and what outcome to improve. In contrast, a Mechanism describes computational operations and their associated Effects. To make Problems and Mechanisms directly comparable, MCG reformulates each Problem into a contextual Requirement and each Mechanism into an Articulation that links its operation with its source-supported Effects (Section 4.3).

To quantify how effectively MCG bridges this representational gap, we encoded Problems, Requirements, Mechanisms, and Articulations with MPNet [35], and visualized them in a shared

two-dimensional projection (Fig. 4a,b). Raw Problems and Mechanisms occupied largely separate regions, with a density overlap of 0.131 in the projection and a normalized centroid gap of 0.459 in the original 768-dimensional space. This separation reflects a difference in how the two are written: Problems typically focus on domain-specific application bottlenecks, whereas Mechanisms describe low-level mathematical operations. By reformulating Problems into Requirements and Mechanisms into operation-effect Articulations, MCG aligns these heterogeneous descriptions into a common semantic format. After reformulation, the density overlap between Requirements and Articulations rose to 0.722 and the normalized centroid gap narrowed to 0.316. Articulations also remained close to their source Mechanisms. When each of the 5,705 Articulations was evaluated against its source Mechanism and a fixed background set of 1,024 randomly sampled Mechanisms, an Articulation maintained a higher cosine similarity to its source Mechanism than to 97.6% of the background items on average. These measurements indicate that reformulation brings Problems and Mechanisms into a comparable form while retaining Mechanism-specific content, which allows direct semantic matching.

We next tested whether MCG enables the reuse of methodological reasoning stored in the RPKG as prior knowledge for new problems. To evaluate whether MCG can more effectively retrieve historical Mechanisms that are suitable as prior knowledge for target Problems, we conducted a retrospective evaluation across 2,429 Problems published between 2020 and 2025. To limit temporal leakage, candidate Mechanisms for each target Problem were restricted strictly to those published prior to the target Problem's publication year, with the ground-truth Mechanism withheld during ranking. For each Problem, we constructed a unified candidate pool by merging and deduplicating the top 10 Mechanisms retrieved by seven first-stage retrieval systems, yielding a fixed set of historical candidates. To determine whether a candidate corresponded to the Mechanism used in a later published method, candidates were rated against later emerging Mechanisms using GPT-5.5 as an LLM judge with a carefully designed prompt on a four-level scale (Section 4.5.3). To ensure the reliability of this automated assessment, we validated the judge's ratings against independent human annotations across 300 sampled Problem–Mechanism pairs. The LLM judge demonstrated strong agreement with human evaluators (ρ = 0.893, quadratic-weighted κ = 0.824), exact agreement on the four-level grade was lower (approximately 60%), with most disagreements between adjacent grades, supporting the use of the automated ratings as the reference for evaluation.

Across retrospective evaluations, MCG ranked above all eight comparison conditions in both AUROC and Hit@10, including lexical, latent-semantic, graph-based, and popularity approaches (Fig. 4c–e). First, when scoring the exact same candidate pool with the MedCPT [36] cross-encoder, MCG improved query-macro AUROC from 0.743 for raw Problem–Mechanism descriptions to 0.783 under reformulation (Fig. 4c). Because the candidate pool and scoring model were unchanged, this gain is attributable to Requirement–Articulation reformulation. In an ablation without the cross-encoder, MPNet-based scoring similarly improved from 0.637 to 0.715, indicating that the benefit of reformulation was not specific to the cross-encoder. Second, MCG had the highest AUROC in every publication year from 2020 to 2025, with annual values between 0.772 and 0.794 (Fig. 4d), indicating that the gain was not confined to particular years. Third, for shortlist construction, MCG placed at least one retrospective match among the top 10 candidates for 69.1% of target Problems, compared to 63.9% for the control using raw descriptions (Fig. 4e). Thus, these retrospective results suggest that reformulation improves the ranking of historical Mechanisms that correspond to later solutions, supporting their use as prior knowledge for hypothesis generation.

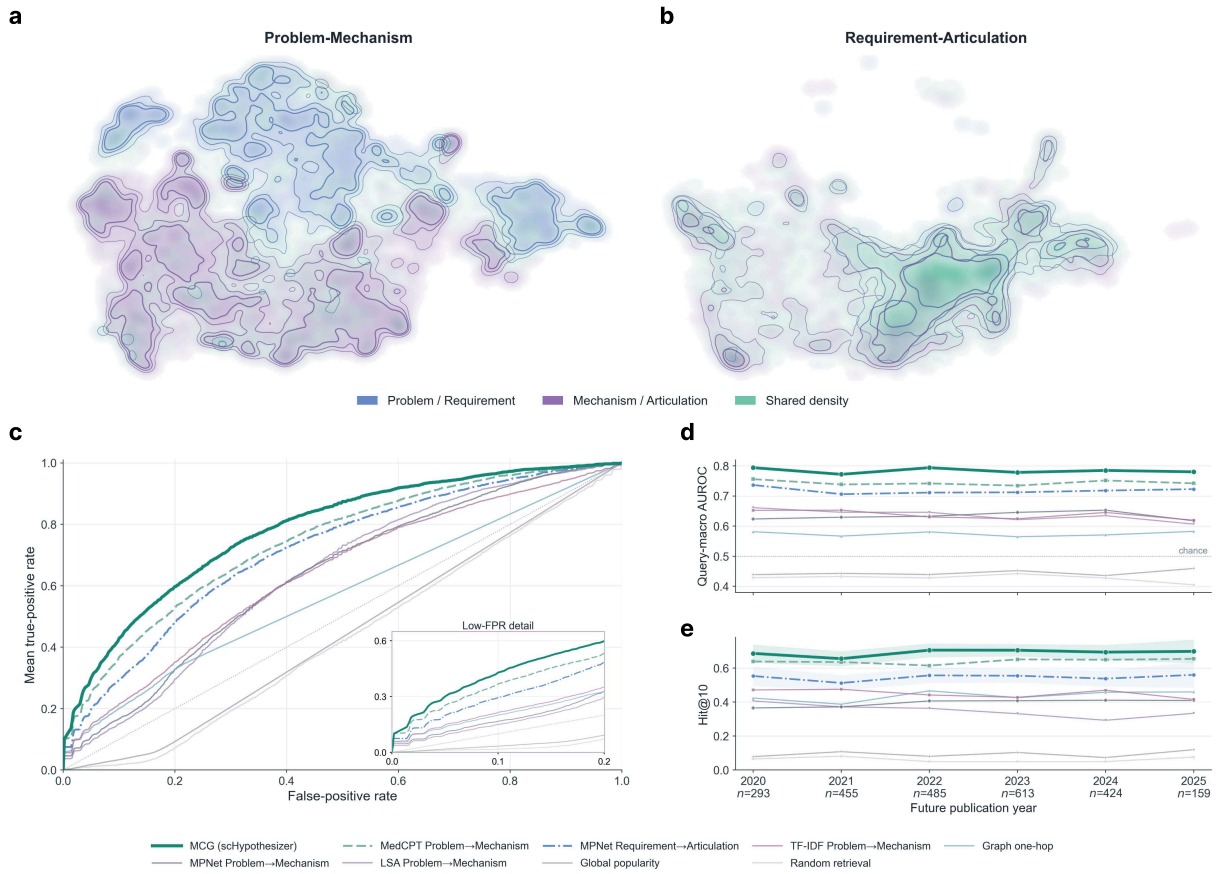


Figure 4 | Causal reformulation and retrospective retrieval of historical Mechanisms. (a) Density contours of raw Problem and Mechanism descriptions, encoded with MPNet and projected to two dimensions by PCA followed by UMAP. **(b)** Requirement and Articulation embeddings mapped through the same fitted projection. **(c)** Receiver-operating-characteristic curves averaged equally across the 2,429 Problems, using the same assessed candidate set for each Problem across methods. The inset enlarges the low-false-positive-rate region. **(d)** AUROC averaged across Problems, by Problem publication year. **(e)** Hit@10 by Problem publication year; *n* denotes the number of evaluated Problems in each year.

2.4. RPKG-derived priors shape candidate hypothesis formulation

Following the retrieval of historical Mechanisms, scHypothesizer's Proposer agent utilizes each retained Problem–Mechanism pair, alongside its supporting RPKG evidence, as a causal anchor to formulate initial method hypotheses. These raw candidates are then iteratively refined through a coordinated Proposer-Critic agent workflow. To examine how RPKG-derived priors and agent coordination shape hypothesis formulation, we evaluated 30 method hypotheses generated for unresolved problems using the RPKGs previously constructed across the three topics, comprising 10 method hypotheses each for Ambient, Ref. map, and Integration.

We first investigated how grounding in RPKG Mechanisms influences the structural architecture of generated hypotheses. Specifically, we compared pre-critic MCG hypotheses against a simple RAG baseline under matched generation conditions, where simple RAG retrieved literature references solely by title similarity to the target Problem. We compared the two candidates for each of the 30 Problems (Section 4.5.6), . The two conditions differed in how hypotheses were constructed:

all 30 MCG hypotheses preserved the RPKG-derived Mechanism as their central causal operation, organizing the proposed workflow around this foundational prior. In contrast, 23 of 30 simple RAG hypotheses resulted in decentralized module stacking, simply concatenating sub-components from multiple papers without an anchoring causal operation (occurring in 9/10 Ambient, 7/10 Ref. map, and 7/10 Integration Problems). LLM quality ratings did not distinguish the two conditions (6.87 versus 6.80; Supplementary Table S6), so this structural difference was not reflected in LLM-assigned scores.

We next examined whether the integration of RPKG knowledge grounding and agentic refinement translates into researcher preference for downstream method development. To assess whether fully refined scHypothesizer hypotheses were preferred over less-structured alternatives, we conducted a randomized choice experiment comparing fully refined scHypothesizer hypotheses against two controls across all 30 Problems: an Abstract-only control (providing raw paper abstracts instead of RPKG Mechanisms) and a Problem-only control (generated solely from the Problem description without retrieved evidence; Methods). Candidate hypotheses were anonymized, matched in length and style, and evaluated by six researchers with single-cell analysis experience. Across 180 total blind decisions, researchers selected the scHypothesizer candidate in 54.4% of cases (98/180), compared with 25.0% (45/180) for the Abstract-only control and 20.6% (37/180) for the Problem-only control (Supplementary Table S1 and Supplementary Data 3). Researchers thus preferred scHypothesizer candidates more often than either control.

2.5. scHypothesizer provides informative evidence through early empirical assessment

To evaluate whether written method hypotheses translate into verifiable algorithmic effects, scHypothesizer assists researchers by coordinating early empirical assessments aligned with each hypothesis's central Mechanism and intended Effect. These early assessments serve as empirical probes to determine whether proposed mechanisms hold in practice, identify their operational boundaries, and expose inherent trade-offs. We selected four researcher-approved candidate hypotheses spanning the Ambient, Ref. map, and Integration topics and evaluated their automated prototypes alongside established task-specific baselines. For each candidate, we constructed dual evaluation suites comprising a General performance board measuring standard task success and a Targeted capability board explicitly isolating the specific algorithmic advance proposed by the hypothesis (Supplementary Note 2 and Supplementary Table S3). Aggregated board scores (Fig. 6) showed setting-dependent benefits in two cases, a trade-off in one, and one hypothesis whose central claim was not supported.

AmbientCap (Fig. 5a). The assessment identified a rare-cell-preservation benefit alongside differences in marker retention across settings. Designed to remove ambient RNA while preserving rare-population markers, AmbientCap ranked first on the rare-cell-preservation board, with a score of 0.97. Its rare-cell nearest-neighbor overlap on PBMC was 0.4353, compared with 0.1238 for the strongest baseline. It also led this measure and cell-type silhouette across all four targeted cohorts. On the General performance board, AmbientCap ranked second with a score of 0.67. Kidney contamination root-mean-square error was 0.1207, the lowest among the compared methods, followed by 0.1340 for the next method. However, marker retention was weaker on single-nucleus and broad-label cohorts. On LungMAP, for example, marker retention was 0.9235, compared with approximately 0.99 for the strongest comparator. The resulting profile distinguished the targeted preservation benefit in the evaluated droplet settings from reduced marker retention in other cohorts.

scOSCAR (Fig. 5b). The assessment identified a trade-off between calibrated coverage and novel-cell detection. scOSCAR was designed to return set-valued decisions whose coverage holds under donor and study shift, whereas established point mappers flag novel cells through native or threshold-based rejection without returning prediction sets. On the novel-cell detection board, scOSCAR ranked

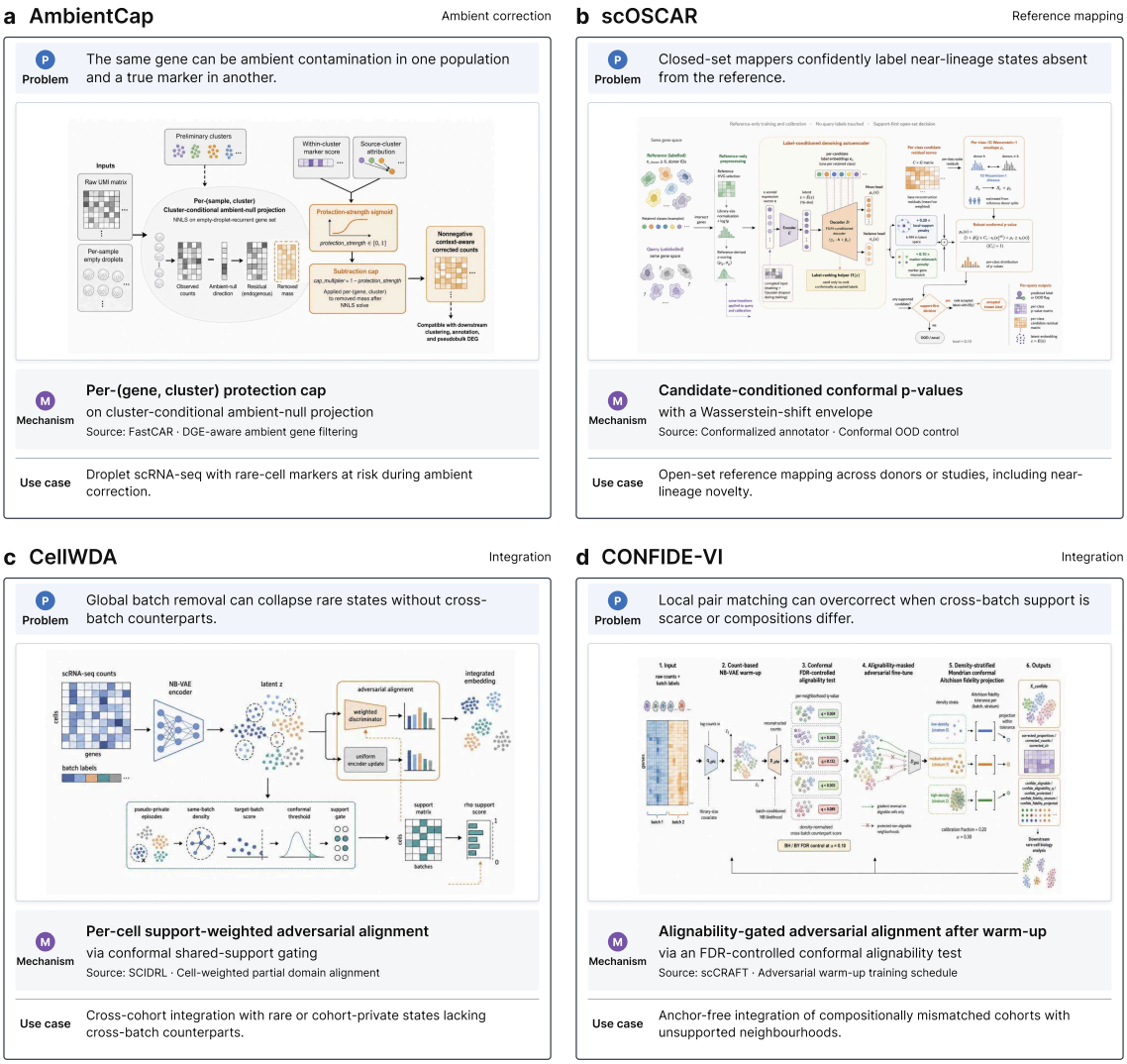


Figure 5 | Method hypotheses advanced to early Empirical Assessment. The target Problem, proposed Mechanism, literature grounding and intended application setting for **(a)** AmbientCap, **(b)** scOSCAR, **(c)** CellWDA and **(d)** CONFIDE-VI, respectively.

fifth of ten methods with a score of 0.73, below four established mappers (0.76–0.87) and above the three alternative generation conditions. Detection was dataset-dependent: against scANVI with a confidence threshold, UNKNOWN F1 was higher on newborn lung (0.844 versus 0.618), slightly lower on liver (0.269 versus 0.311), and lower on kidney (0.279 versus 0.751), where held-out types lay close to retained lineages. On the coverage-under-shift board, scOSCAR ranked first with a score of 0.74, compared with 0.26 for CSA, the only established method that returns prediction sets; coverage errors were 0.0272 on LungMAP and 0.0097 on retina. On the General performance board, scOSCAR ranked fourth with a score of 0.82, and its known-cell macro-F1 was 0.936, compared with 0.980 for the strongest mapper. The resulting profile distinguished calibrated coverage, which point mappers do not provide, from novel-cell detection, which did not exceed that of established mappers and was weakest for near-lineage novelty.

CellWDA (Fig. 5c). Selective batch alignment showed benefits in the evaluated partial-overlap settings. By restricting alignment to cells with estimated cross-batch support, CellWDA ranked first

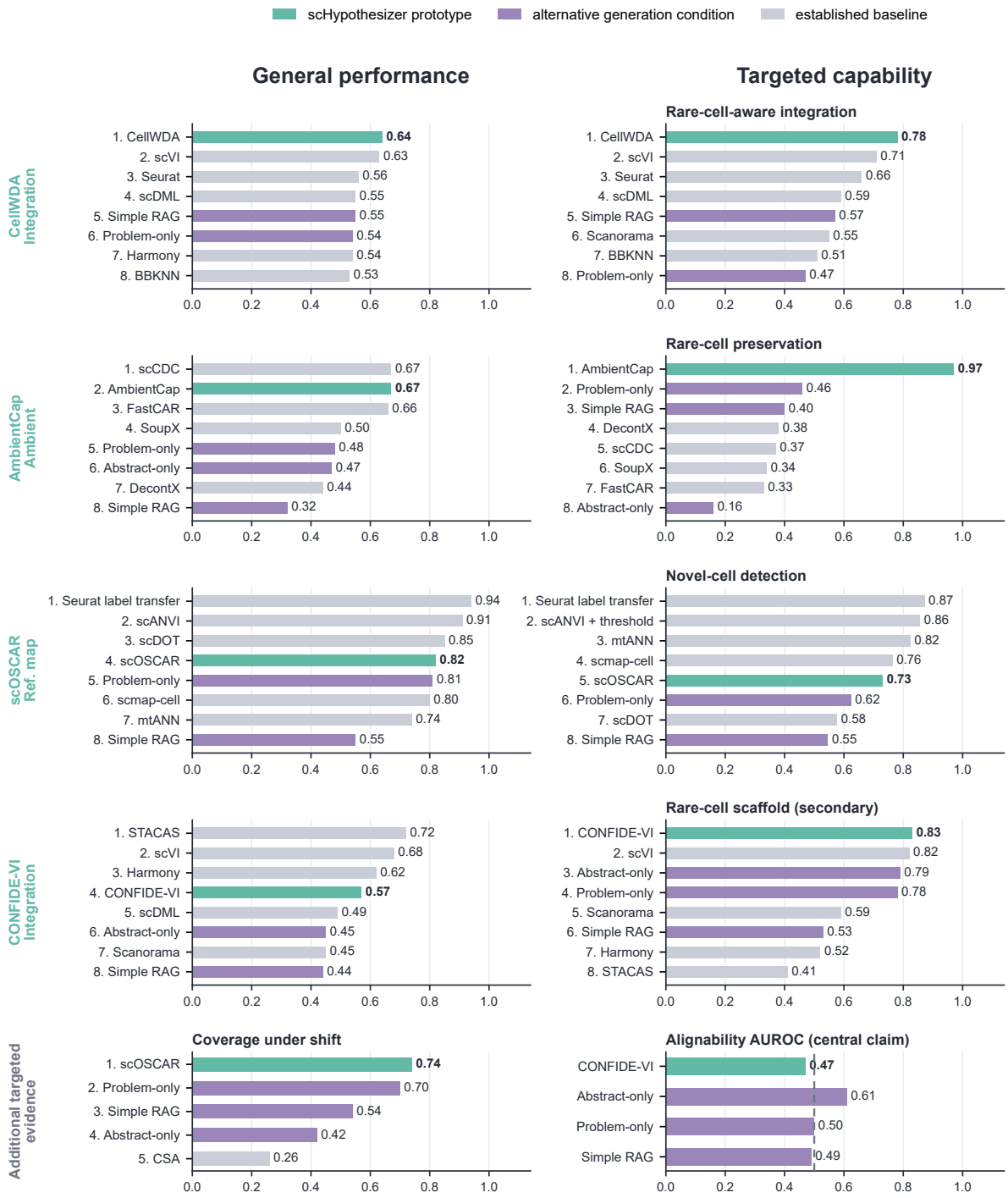


Figure 6 | Early empirical assessment of four method hypotheses. General performance and Targeted capability boards are shown for each prototype. The lower panels report additional targeted evidence: coverage under shift for scOSCAR and the central alignability claim for CONFIDE-VI on its native AUROC scale. Green marks the assessed prototype, purple marks less-structured generation conditions, and gray marks established baselines. The dashed line denotes chance alignability performance. Component metrics were normalized within dataset–task–metric combinations before aggregation; board scores are case-specific and should not be compared across tasks. Displayed values are rounded to two decimals; ties in the displayed values are ordered by the unrounded scores. Each panel shows up to the eight highest-ranked methods.

on the Targeted capability board with a score of 0.78. Mean shared-support AUROC was 0.836, compared with 0.809 for the next baseline. Its calibrated abstention set also recovered 88.8% of documented rare cells, compared with approximately 80% for the closest analog. These targeted results were accompanied by a first-place General performance score of 0.64 and graph connectivity of 0.968, although clustering performance remained intermediate. The donor-only, single-chemistry cohort showed a different pattern: CellWDA recovered 80.0% of donor-confined cells, compared with 98.9% for the baseline. Thus, the assessment connected the selective-alignment benefit to the partial-overlap setting and identified a setting in which the heuristic recovered more donor-confined cells.

CONFIDE-VI (Fig. 5d). CONFIDE-VI combined an adversarial warm-up schedule from the RPKG with a conformal alignability test that decides where alignment is applied. Its central claim, that this test separates alignable from unalignable cells, was not supported by the empirical assessment. Mean alignability AUROC was 0.47, close to the 0.50 chance level. Top-ranked rare-marker recovery and log-fold-change sign concordance were also lower than their corresponding comparisons by 0.073 and 0.309, respectively. On the General performance board, the prototype ranked fourth with a score of 0.57. Secondary measurements nevertheless revealed more limited signals: the rare-cell-scaffold composite was 0.83. This composite combined false mixing among unalignable cells, rare-state nearest-neighbor purity, and rare-state trade-off AUROC. Density-stratified outputs also yielded low-density marker-retention ratios of 1.665 in pancreas and 1.057 in PBMC-SCA. Reporting these measurements separately kept the rare-cell results distinct from the unsupported central alignability claim.

Across these empirical cases, early empirical assessment provided structured evidence directly tied to each underlying hypothesis. For AmbientCap and CellWDA, the assessments supported the targeted Effects in specific settings and identified the dataset characteristics under which these advantages diminished. For scOSCAR, the assessment identified a trade-off between calibrated coverage, which was supported, and novel-cell detection, which did not exceed that of established mappers. For CONFIDE-VI, the assessment did not support its central alignability claim, while secondary measurements indicated signals that could inform redesign. Rather than serving as a standard performance ranking, early empirical probing connects abstract mechanisms with actionable evidence regarding their intended Effects, operational boundaries, and failure modes.

To assist researchers in comprehensively evaluating these findings, scHypothesizer compiles the proposed model architecture, experimental scheme, empirical results, and diagnostic conclusions into a manuscript-formatted research record. To assess the utility and informativeness of these records, we submitted them to the same six researchers for systematic review (Supplementary Note 6 and Supplementary Table S7). Reviewers mapped the empirical strengths, performance boundaries, and failure conditions directly back to the underlying RPKG Mechanisms and judged the records informative for deciding whether to refine, pivot, or discontinue each hypothesis. Overall, these cases illustrate how scHypothesizer can return claim-aligned empirical evidence to inform researchers' decisions about further development.

3. Discussion

In this study, we investigated the use of coordinated agents to preserve and reuse methodological reasoning as prior knowledge for computational method development in single-cell RNA sequencing. We then presented scHypothesizer, a knowledge-grounded agentic framework that operationalizes this concept by structuring literature-derived knowledge into Research Primitive Knowledge Graphs (RPKGs). The system explicitly connects computational *Problems*, underlying *Mechanisms*, and

intended *Effects* with article-level source provenance. Our empirical evaluations indicate that explicitly decoupling and matching problem requirements against candidate articulations improves mechanism selection compared to raw text retrieval. Furthermore, in retrospective temporal evaluation, the framework prioritized historical mechanisms that corresponded to those used in later publications, supporting their use as prior knowledge for hypothesis formulation.

The primary value of scHypothesizer lies in bridging literature-derived prior knowledge with early empirical assessment. The coordinated agents operationalize extracted mechanisms as the functional core of new algorithmic proposals. Through targeted empirical assessments, the framework systematically evaluates whether intended effects materialize in practice, identifies setting-dependent trade-offs, and exposes unsupported claims before substantial engineering investment. Reviewers judged the resulting research records informative for deciding whether to continue developing a hypothesis.

Several limitations constrain the scope of our current findings. First, our empirical evaluations were restricted to single-modality scRNA-seq count matrices and a limited range of analytical topics. Second, the coverage and fidelity of the RPKG remain bounded by full-text accessibility and extraction accuracy. Methodologically, the temporal evaluation assessed alignment within a frozen candidate pool instead of evaluating the complete spectrum of viable scientific mechanisms. In addition, early empirical assessment was carried out for four researcher-approved hypotheses, whereas the Problem–Mechanism pairings available in an RPKG are far more numerous; these cases therefore illustrate the assessment process and do not characterize how often generated hypotheses are supported. Finally, the current workflow relies on long-context LLMs without an explicit long-term memory architecture, leaving its performance under constrained context windows to be further investigated.

More broadly, this work highlights the potential of coordinated agents to make literature-derived methodological knowledge systematically reusable. A published method contributes not only a specific implementation, but also structured reasoning connecting its Problem, Mechanism, and intended Effect. Preserving this reasoning as source-traceable prior knowledge enables researchers to apply past insights to new computational challenges. Beyond single-cell analysis, extending this framework to other domains could foster the cross-disciplinary reuse of methodological ideas.

4. Methods

This section describes the methodology underlying scHypothesizer (Fig. 1) and the evaluation protocols used to assess its outputs. We first define the system architecture and shared agent execution principles and then follow the three fixed stages in sequence. For RPKG Construction, we specify our proposed RPKG schema, literature retrieval and primitive-aware distillation, and the relation-verification and primitive-normalization procedures used to assemble each RPKG. For Hypothesis Generation, we describe Mechanism-centered Generation (MCG) and the Proposer-Critic loop that develops a selected pairing into a method hypothesis. For Empirical Assessment, we describe benchmark planning, controlled execution, and preparation of an assessment record. Finally, we give the protocols for RPKG audit, mechanism-family analysis, causal-representation analysis, retrospective temporal retrieval, blinded human preference, empirical assessment scoring, and review-style assessment of the completed records.

4.1. System architecture and agent execution pattern

scHypothesizer operates as a three-stage coordinated-agent framework driven by a user-defined research topic (Fig. 1a). The framework executes in a sequential pipeline comprising graph construc-

tion, hypothesis generation, and empirical assessment. In this framework, each stage consumes the artifacts generated by its predecessor to produce an inspectable research record. All durable state, including retrieved records, extracted graph primitives, execution code, and manuscript assets, is maintained in a shared external workspace.

Within each stage, agents operate under a unified plan-then-execute pattern that decouples high-level reasoning from external interactions [37–41], consistent with recent scientific agent architectures [12, 13, 15, 16]. Before taking action, an agent generates a concise stage-local plan, issues bounded tool calls, validates the outputs, and logs the results into the shared workspace. Unless noted otherwise, all agents are powered by a general-purpose GPT-5.4 backend and specialized solely through stage-specific instructions, output schemas, and restricted tool sets. The Critic agent uses Gemini 3.1 Pro, while cross-LLM extraction experiments incorporated GPT-5.4, Claude Opus 4.7, and Qwen 3.7-Max.

Low-level operations are exposed through atomic, stateless Model Context Protocol (MCP) interfaces [42], where each interface performs a discrete task such as querying PubMed, downloading articles, or accessing the knowledge graph. To maintain strict operational boundaries, tool access is partitioned by stage. Specifically, RPKG Construction agents have access to retrieval, parsing, summarization, and graph-building tools; Hypothesis Generation agents are restricted to graph-retrieval, ranking, proposal-formulation, and critique tools; and Empirical Assessment agents are granted access to code-execution environments, reference-checking, and report-assembly tools. This strict scoping ensures that each agent operates exclusively within its designated functional role.

4.2. RPKG Construction

4.2.1. RPKG schema

The first stage constructs RPKGs based on a predefined schema: a directed, typed multigraph with six node types and a fixed relation vocabulary (Fig. 1b). Predefining this schema ensures that extracted literature evidence enters downstream generation as structured, typed entities and relations within a unified vocabulary. This design aligns with schema-constrained scientific information extraction, where entity and relation inventories are specified prior to graph construction [24–27]. Each extracted primitive retains a textual description and source-paper provenance linked to its corresponding summary.

The schema defines one source node type and five domain primitives. A Paper node serves as the bibliographic container for all extracted evidence. Among the primitives, a Method node represents a computational algorithm, statistical model, or processing pipeline, whereas a Problem node encapsulates an analytical limitation or need that motivates method development. A Mechanism node captures a specific algorithmic component introduced to address a problem, named for its primary intended effect (for example, adversarial alignment for batch correction). Finally, an Effect node denotes a measured or claimed consequence of a mechanism, and a Task node specifies the analytical setting in which that effect operates.

Edges represent either structural or functional method-development relations. Structural relations record lineage and provenance, including contains (method to component), published_in (method to paper), built_on (method to prior method), and compared_to (method to baseline). Functional relations encode the core reasoning required for hypothesis generation: motivates (problem to mechanism), addresses (method to problem), incorporates (method to mechanism), achieves (mechanism to effect), enables (effect to task), and limits (problem to method). Chaining these functional relations yields the canonical reasoning path used for graph quality assessment (Fig. 1b):

$$\text{Problem} \xrightarrow{\text{motivates}} \text{Mechanism} \xrightarrow{\text{achieves}} \text{Effect} \xrightarrow{\text{enables}} \text{Task} \tag{1}$$

4.2.2. Literature retrieval, full-text preparation, and primitive-aware distillation

To populate this schema, the RPKG Construction stage (Fig. 1c) first assembled an open-access full-text corpus for a user-specified topic. This process followed four sequential steps: query expansion, title and abstract screening, open-access verification, and corpus collection and parsing.

Query expansion initiated corpus collection. The Retrieval Agent (Fig. 1c) generated two seed queries based on the target topic, executed the initial retrieval, and used the retrieved abstracts as feedback to refine the query set for higher relevance and broader coverage. The refined query set captured alternative names for the task, problem, algorithmic family, and evaluation setting.

Before downloading any article, the Retrieval Agent screened each returned record using only its title and abstract, returning a binary selection decision alongside a brief rationale. The screening rule was recall-biased. Articles describing computational methods, benchmarks, or resources relevant to algorithm development were retained. Articles primarily addressing biological applications, clinical studies, laboratory protocols without a computational contribution, or unrelated domains were excluded.

Each retained article was then verified against the PubMed Central (PMC) Open Access (OA) file lists or queried through the OA Web Service to confirm its licensing metadata and downloadable package availability [43, 44]. Articles lacking eligible open-access packages were excluded. The PubMed free-full-text filter served solely as an initial candidate pool and was never treated as download authorization. No publisher-site web scraping or non-OA retrieval was used. Full texts were retrieved exclusively from the PMC OA subset in accordance with their licenses. The system stored only derived artifacts—specifically summaries, graph primitives, and verified relations—and did not retain the original full-text bodies.

Finally, during corpus collection and parsing, each downloaded JATS/NXML article was parsed into a structured representation that preserved its metadata, license, abstract, section hierarchy, text, figures, tables, formulas, and references. Guided by the RPKG schema, the Distillation Agent (Fig. 1c) processed each parsed article to generate two targeted summaries: an article summary capturing the full methodological scope for graph construction, and a novelty summary isolating original contributions for both graph construction and subsequent hypothesis generation. These structured summaries served as the primary evidence inputs for primitive extraction and relation scoring.

4.2.3. Primitive extraction and graph consolidation

From the article summaries, we constructed the graph through typed, schema-constrained extraction, relation verification and primitive normalization, retaining source provenance throughout (Fig. 1c). The Schematization Agent extracted candidate primitive nodes and typed relations from each summary according to the fixed RPKG schema. Each candidate node included its type, name, description, source-paper provenance and supporting evidence. Candidates were then pooled across articles (Supplementary Note 5).

The Relation Verification Agent assessed each candidate relation against its source summary and assigned a score between 0 and 1 based on textual support and topical relevance. Relations scoring below 0.6 were discarded. The Normalization Agent then consolidated duplicate or synonymous candidate nodes using Algorithm 1.

Normalization began by standardizing node names using Unicode NFKC normalization, whitespace normalization, and case folding. Candidate nodes with identical standardized names were grouped within each primitive type. Removing non-alphanumeric characters from the standardized names

produced alphanumeric skeletons. Same-type clusters with matching skeletons were submitted to the Normalization Agent for review; a matching skeleton alone did not authorize a merge.

For semantic comparisons, each candidate node's primitive type, name, and attributes were serialized and encoded using MPNet. Let $\mathbf{e}_i$ denote the embedding of candidate node i, and n_i its occurrence count. We denote the current set of clusters by C, with each cluster $g \in C$ containing nodes of one primitive type. The cluster embedding $\mathbf{z}_g$ was the ℓ_2-normalized, occurrence-weighted sum of its member embeddings, as defined in Algorithm 1.

A hierarchical navigable small world (HNSW) index [45] retrieved up to $k = 50$ approximate nearest neighbors for each cluster within the same primitive type. Candidate pairs were retained if their cosine similarity was at least $\gamma = 0.80$ and they had neither been reviewed nor been blocked by a previous distinct-entity judgment. Among the remaining candidates, we selected pairs in which each cluster was the other's highest-scoring eligible partner, with ties resolved by cluster identifier. These are referred to as mutual-best pairs. In the algorithm, $\mathcal{P}$ denotes the candidate set of cluster pairs (g, h), where $g, h \in C$.

Lexical and semantic candidates followed the same review and merge rules. The Normalization Agent reviewed the contents of both clusters and assigned one of three labels: SAME ENTITY, RELATED BUT DISTINCT, or UNCERTAIN. A SAME ENTITY pair was merged only if no existing cannot-link constraint prohibited the merge. A RELATED BUT DISTINCT judgment established a cannot-link constraint, preventing the two clusters from being merged. An UNCERTAIN judgment left the pair unmerged without adding a cannot-link constraint.

Pairs were reviewed in batches, with no cluster appearing more than once within a batch. After each batch, merged clusters retained all aliases, source provenance, occurrence attributes, and incident relations. Merge constraints and review history were remapped to the updated clusters, and cluster embeddings were recomputed. Semantic candidate retrieval and review continued until no eligible, unreviewed mutual-best pair remained. A round with no accepted merge did not terminate the procedure if eligible unreviewed pairs remained. Each cluster's canonical name was the member name with the highest total occurrence count, with ties resolved by choosing the longer name and then the lexicographically first name. HNSW supplied candidates for review; semantic similarity alone did not authorize a merge.

The canonical nodes and verified relations were assembled into the final RPKG, a directed, typed multigraph retaining relation confidence scores, source provenance and evidence summaries. Isolated nodes were removed before export, so the reported graph statistics refer to non-isolated method-development primitives.

4.3. Hypothesis generation through Mechanism-centered Generation

Given the RPKG, MCG develops candidate method hypotheses by pairing Problems with source-supported Mechanisms and using the linked literature evidence to elaborate each pairing. The procedure comprises reformulation (Fig. 7a), candidate construction and ranking (Fig. 7b), and hypothesis refinement (Fig. 7c). Building on literature-based discovery with concept pairs [26, 46], MCG pairs typed nodes with graph provenance. Each Problem node specifies a target problem. Each canonical Mechanism and its verified outgoing achieves relations supply a candidate operation together with all Effects supported by its evidence.

Reformulation begins by enumerating the canonical Mechanisms in the RPKG and collecting all verified Effects attached to each Mechanism. Because articulating every record and scoring all possible pairs incurs model cost, candidates are ranked by their verified graph support and limited to the

Algorithm 1 Typed primitive normalization

Require: Candidate primitive nodes, MPNet encoder, Normalization Agent; $k = 50$, $\gamma = 0.80$
Ensure: Canonical primitive nodes with aliases, attributes, provenance and relations

Standardize node names as described above.
Encode each node's type, name, and attributes with MPNet to obtain $\mathbf{e}_i$.
$C \leftarrow$ clusters of exact-name matches within each primitive type.
$\mathcal{P} \leftarrow$ same-type cluster pairs with matching alphanumeric skeletons.
Review $\mathcal{P}$ in batches and update C using the review and merge rules above.
while true do
    Compute the current cluster embeddings:
    $$\mathbf{z}_g \leftarrow \frac{\sum_{i \in g} n_i \mathbf{e}_i}{\left\| \sum_{i \in g} n_i \mathbf{e}_i \right\|_2}, \qquad g \in C.$$
    Retrieve up to k same-type neighbors per cluster using HNSW.
    $\mathcal{P} \leftarrow$ distinct cluster pairs from the retrieved neighbors.
    Retain pairs $(g, h) \in \mathcal{P}$ with $\mathbf{z}_g^\top \mathbf{z}_h \geq \gamma$.
    Remove blocked and previously reviewed pairs from $\mathcal{P}$.
    Retain mutual-best pairs, resolving tied scores by cluster identifier.
    if $\mathcal{P} = \varnothing$ **then**
        break
    end if
    Review $\mathcal{P}$ in batches and apply the merge rules to C.
    Update cluster contents, embeddings, constraints, and review history after each batch.
end while
Select each cluster's canonical name using the occurrence counts and tie-breaking rules above.
return Canonical primitive nodes obtained from C.

top $K = 80$ before articulation, following retrieval-augmented generation and confidence-filtered knowledge-graph construction [25, 47]. This cap applies to topic-specific generation only; in the domain-scale retrospective evaluation, all eligible historical Mechanisms were articulated.

The retained graph records are then rendered as complete causal texts. For each Problem, a language model produces one mechanism-agnostic Requirement that preserves the analytical setting, failure pathway, and desired outcome while prohibiting the name or proposal of a solution mechanism. The input comprises the complete Problem description and its eligible graph-neighbor evidence. For each canonical Mechanism, the model produces one Articulation describing the causal operation, the object on which it acts, and all Effects supported by the linked evidence. Candidates with insufficient or contradictory evidence for a coherent Articulation, and Problems too vague to yield an actionable Requirement, are excluded. The Requirement and Articulation instructions are reproduced in Supplementary Note 5.

For semantic matching, the MedCPT cross-encoder jointly encodes each complete Requirement–Articulation text pair and returns a scalar relevance score [36]. Because a cross-encoder returns a pairwise relevance score rather than independent embeddings, this ranking does not define a separate embedding space. For each Problem, the three highest-scoring candidates are retained after a graph-level novelty filter. This filter removes pairings for which an existing Method already addresses the Problem using the same Mechanism, while retaining transfers in which the Problem, analysis setting, or intended Effect differs. Each surviving pairing therefore links a Problem to a source-supported Mechanism not already instantiated for that Problem in the topic-specific graph.

The remaining pairings then pass through a causal-fit gate that decides whether they are retained.

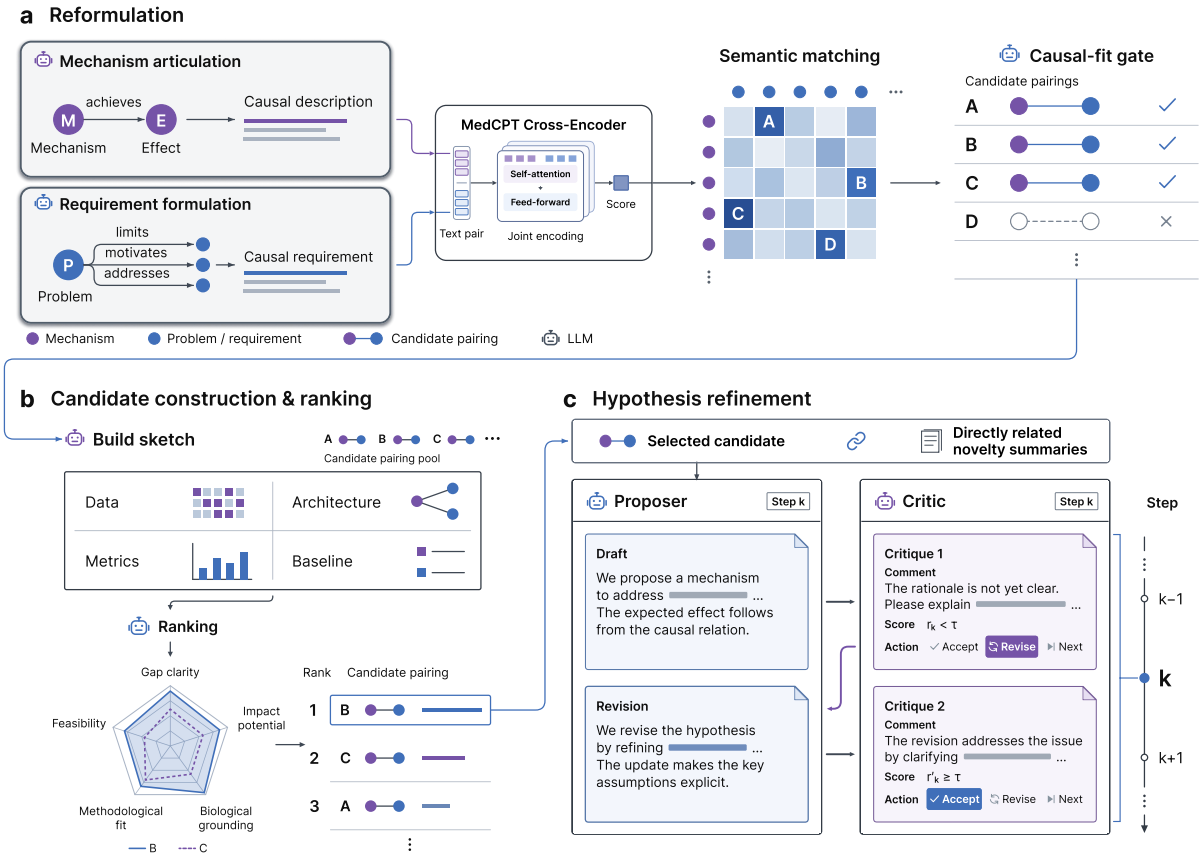


Figure 7 | The MCG procedure. (a) Reformulation of Problem and Mechanism through Mechanism articulation, Requirement formulation, MedCPT pairwise scoring, and a causal-fit gate. **(b)** Candidate construction and ranking using build sketches and multiple review criteria. **(c)** Hypothesis refinement through a Proposer-Critic loop using RPKG evidence and directly related novelty summaries. At refinement step k, r_k and r'_k denote a criterion score before and after revision, respectively, and τ is its acceptance threshold. Advancement requires all review criteria to pass.

For each pairing, the LLM judges whether the mechanism can intervene on the problem, instead of merely sharing vocabulary with it or serving as a diagnostic or benchmark. Only interventional fits proceed to candidate construction and ranking (Fig. 7b).

Each retained pairing is expanded into a build sketch. The sketch is seeded with the articulated mechanism description, so its proposed components are derived from the recorded causal mechanism rather than from the mechanism name alone. It specifies the prototype name, data assumptions, algorithmic components, loss or objective functions, and an initial empirical assessment plan. The candidate pairings are then ranked on five dimensions: gap clarity, impact potential, feasibility, biological grounding, and methodological fit (Fig. 7b). A critically weak score on any dimension caps the overall rank, so strength on another dimension cannot compensate for it. This ranking determines the order in which pairings enter hypothesis refinement. The operative decision rules are reproduced in Supplementary Note 5.

Each candidate pairing is refined into a method hypothesis through a stepwise Proposer-Critic loop (Fig. 7c), following feedback-driven refinement in LLM systems [48, 49]. To initiate each proposal, the Proposer uses the candidate pairing, its supporting RPKG evidence and the novelty summaries

of papers directly linked to the matched Mechanism. The Critic uses a separate language-model backend, ensuring that each proposal is reviewed by a model different from the one that generated it. The key proposer and critic instructions are reproduced in Supplementary Note 5.

The proposal then progresses through three gated stages, each allowing up to five revision rounds. First, the Proposer frames the direction by defining the analysis scope and biological challenges that the method has to address and by committing to a single testable claim describing what the method would demonstrate and why existing methods could not achieve it. The direction review scores the proposal on six criteria (Supplementary Note 5) and must approve this framing before the method is developed further. Next, the Proposer uses the linked summaries and RPKG paths to identify the limitations and gaps underlying the claim and to formulate a method hypothesis from them. The mechanism review, conducted by four parallel specialists, assesses biological grounding, technical novelty, feasibility, and evaluability. Finally, the Proposer drafts the evaluation strategy, including the task types, baseline families, metrics, and ablations required to test the claim. The benchmark review assesses appropriateness, baseline fairness, metric completeness, coherence, and unique advantage.

At each review, the Critic scores the relevant output on a 1–10 scale. A proposal advances only when every required dimension reaches the acceptance threshold, $\tau = 8$. Otherwise, the Proposer revises it using the written feedback, or the candidate is removed from the internal queue. In Fig. 7c, k indexes the three refinement stages described above; the illustrated scores refer to an individual review criterion. These routing states control refinement and do not authorize empirical work. Retained outputs are consolidated into method hypotheses for researcher consideration. The 30 records used in the formulation study are reproduced in Supplementary Note 4 and Supplementary Data 1. Four internally ranked candidates advanced to empirical assessment with researcher approval: one Ambient candidate, one Ref. map candidate, and two Integration candidates.

4.4. Empirical Assessment

Empirical Assessment began only after researchers had selected and authorized a candidate. Its role was to operationalize a prespecified claim as a fixed benchmark and return inspectable evidence. The evaluation strategy specified only the types of datasets, baselines, metrics, and ablations required by the claim. Before execution, the Benchmark Planning Agent verified the corresponding resources and converted the evaluation strategy into a fixed, executable benchmark plan. Using the available tools, it queried the literature, public datasets, and method repositories to confirm that the reference papers, candidate datasets, established baselines, and standard metrics existed and were accessible. Based on these verified resources, it fixed the algorithmic formulation, contribution list, datasets, baseline set, metrics, and ablations. It also confirmed that each headline claim was tied to a computable metric and that the falsifying ablation had been specified in advance. A separate Critic first checked contribution focus and verifiability and then assessed baseline fairness and dataset–metric–ablation coverage. Plans that failed these checks were revised within the planning stage or withheld from execution. Only resource-verified, researcher-authorized plans were passed to execution.

Each researcher-authorized, resource-verified plan was passed to an Execution Agent, which implemented and ran the benchmark code. Execution proceeded through four phases in a fixed order: data acquisition, prototype implementation, baseline execution under a shared protocol, and evaluation of the planned metrics. Three conventions kept the methods comparable across phases. First, data acquisition converted every dataset into a common HDF5-backed AnnData (`h5ad`) representation accessible from both Python and R, ensuring that all methods consumed identical inputs. Second, each baseline ran in an isolated virtual environment, allowing Python- and R-based methods to run side by side without package conflicts. Third, a shared metrics script computed every reported metric from each method's outputs, ensuring that all methods were scored identically. Each

phase was self-contained, with its own inputs, generated scripts, and success conditions, and wrote its outputs as assets for use in subsequent phases.

Within each phase, the Execution Agent first inspected the available files and wrote a phase-specific plan without executing code. It then implemented the plan in one or more scripts, executed them, and checked the predefined success conditions for that phase. These conditions were concrete, machine-checkable assertions applied to the phase outputs. For example, during data acquisition, the preparation script reloaded all written files and asserted a fixed set of conditions before exiting. It required every declared input file to load successfully, each input's cell count to match its declared metadata, all hidden labels to remain masked, and repeated execution to produce byte-identical outputs. A failed assertion raised an error and caused the phase to fail. Each phase recorded failed attempts and reusable assets in a persistent history, allowing subsequent phases and retries to avoid repeated dead ends and reuse verified datasets, baselines, and environment setups.

4.4.1. Manuscript-formatted record assembly

The assets generated across RPKG Construction, Hypothesis Generation, and Empirical Assessment were assembled into a manuscript-formatted research record (Fig. 1a). This record directly linked the proposed method hypothesis to its supporting graph evidence, benchmark design, executable code, metrics, tables, and references. To ensure rigor, reference checks flagged uncertain citations for author verification, while consistency checks cross-validated numerical claims against raw experimental logs. The resulting document provided a transparent substrate for human evaluation, explicitly presenting which claims were supported, setting-dependent, or unsupported, while leaving final development decisions entirely to the researcher.

4.5. Evaluation protocols

4.5.1. Mechanism-family vocabulary and topic enrichment

We derived a mechanism-family vocabulary from the 6,046 canonical Mechanism nodes without changing the normalized RPKG. For each node, we collected its canonical name, alternative names, the wording recorded for individual mentions, and the mechanism names and technique descriptions associated with those mentions. We standardized the text using Unicode NFKC normalization and converted it to lowercase. Standard English stop words and numeric tokens separated the text into segments. Within each segment, we extracted all sequences of two to four consecutive words. Each phrase was counted once per canonical node, even if it appeared more than once in the text associated with that node. Thus, a phrase's document frequency was the number of Mechanism nodes containing it.

We ranked candidate phrases using four scores. The first was a non-negative C-value calculated using node counts as phrase frequencies. The second and third were log-odds z-scores calculated with an informative Dirichlet prior. These compared phrase frequencies in Mechanism text with frequencies in the other primitive types and in article titles and abstracts, respectively. The fourth was the logarithm of the number of Mechanism nodes containing the phrase. For each score, we converted the candidate values to percentile ranks, assigning the same percentile to tied values. The final phrase score was the mean of the four percentile ranks, with equal weight given to each.

In the primary setting, phrases had to occur in at least five Mechanism nodes, and we retained the 100 highest-scoring phrases. We assessed the sensitivity of phrase selection to the parameter settings by repeating the procedure with minimum node counts of 3, 5, or 10, maximum phrase lengths of three or four words, and selection sizes of 50, 100, or 150 phrases. These choices gave 18 parameter

combinations. For each phrase, we calculated how often it was selected among the parameter settings in which it was evaluated. We retained phrases selected in at least half of those settings, yielding 98 candidates. Signed Dunning G^2 statistics were also recorded but were not included in the phrase scores.

For each candidate phrase, we provided an LLM with information from every canonical Mechanism node matched by that phrase. This information included the node's names, the wording and attributes recorded for its individual mentions, its relations to other nodes, and the PMC identifiers of its source articles. With temperature set to 0, the model classified each phrase as a core mechanism, a component, a metric, a task or output, a workflow or software term, or a generic fragment. We retained the 62 phrases classified as core mechanisms. In a second call, the model considered all 62 phrases together and grouped phrases describing the same computational operation that could be applied across different problems. Each phrase was assigned to exactly one family, and a family could contain a single phrase. This produced 28 mechanism families. Human reference decisions were not provided to the model, and we did not manually add, remove, or change the selected phrases or their family assignments.

We then identified the papers supporting each family. For each phrase, we collected all source-paper identifiers attached to its matched Mechanism nodes. For each family, we combined the paper sets of its member phrases and counted each paper once. A paper was therefore linked to a family through the provenance of a matched Mechanism node; the family phrase did not have to appear literally in every linked paper. The 28 families covered 828 Mechanism nodes and 593 of the 1,093 reviewed papers.

We used the external scRNA-tools categories to assign research topics to papers, allowing a paper to belong to more than one topic [34]. Let $N = 1{,}093$ be the total number of eligible papers, n_t the number assigned to topic t, K_f the number supporting family f, and k_{tf} the number belonging to both sets. If topic membership and family support were independent, the expected number of papers in their intersection would be E_{tf}. We calculated this expectation and the fold enrichment as

$$E_{tf} = \frac{n_t K_f}{N}, \qquad \mathrm{FE}_{tf} = \frac{k_{tf}}{E_{tf}}. \tag{2}$$

A fold enrichment above one means that more papers supported the family within that topic than expected under independence. We tested for over-representation using the upper tail of the hypergeometric distribution: $p_{tf} = \Pr(X_{tf} \geq k_{tf})$, where $X_{tf} \sim \mathrm{Hypergeometric}(N, K_f, n_t)$. This is the probability of observing at least k_{tf} family-supporting papers when drawing n_t papers without replacement from the full set of N papers, of which K_f support the family. We applied Benjamini–Hochberg correction to all 924 tests covering 33 topics and 28 families, and denoted the adjusted P values by q_{tf}. A topic–family pair was classified as topic-enriched when it was supported by at least three papers, had fold enrichment above one, and had $q_{tf} \leq 0.05$.

For Fig. 3b, we retained topics and families with at least one topic-enriched association. Among all pairs formed by those topics and families, we displayed every pair supported by at least three papers, whether or not it met the enrichment criteria. This gave 81 displayed associations across 11 topics and 14 families, including 20 topic-enriched associations. Marker color indicates $\log_2 \mathrm{FE}_{tf}$, marker area indicates the number of supporting papers k_{tf}, and a dark outline marks a topic-enriched association. Blank cells indicate fewer than three supporting papers. Negative color values indicate fewer supporting papers than expected from the full set of 1,093 papers. We did not test whether this under-representation was statistically significant.

4.5.2. Causal reformulation and representation geometry

For the temporal analysis, MCG reformulated Problems and historical Mechanisms independently before comparing their representations. Each query Problem was reformulated as a single mechanism-agnostic Requirement describing the analytical setting, failure pathway, and desired outcome without specifying a concrete solution. Requirement construction used the complete Problem text and graph-neighbor evidence supported by sources published before the temporal cutoff. The future Mechanism, Effects, article summary, and novelty summary were withheld. Each historical Mechanism was reformulated as an Articulation describing its causal operation and all Effects supported by that historical occurrence.

We encoded the descriptions with the MPNet sentence encoder [35] and ℓ_2-normalized the resulting 768-dimensional embeddings. For visualization, we fitted a 50-component principal component analysis (PCA) transform followed by uniform manifold approximation and projection (UMAP) to the original Problem and Mechanism embeddings. UMAP used cosine distance, 20 neighbors, a minimum distance of 0.1, and random seed 42. The 4,129 Requirement embeddings and 5,705 Articulation embeddings were then mapped through the fitted pipeline without refitting.

For each representation pair, we constructed the two density maps on a common grid covering their joint projection range. Each map was smoothed with a Gaussian kernel of standard deviation $\sigma = 2.2$ grid cells and normalized so that its grid-cell masses summed to one. Let ρ_g and η_g denote the two normalized masses in cell g. We calculated density overlap as

$$\text{Overlap}(\rho, \eta) = \sum_g \min(\rho_g, \eta_g). \tag{3}$$

Greater overlap indicates closer alignment in the projected space.

We also measured separation in the original embedding space. Let $A = \{\mathbf{x}_i\}_{i=1}^{n_A}$ and $B = \{\mathbf{y}_j\}_{j=1}^{n_B}$ denote the two representation roles: Problems and Mechanisms before reformulation, or Requirements and Articulations afterward. Their centroids are $\boldsymbol{\mu}_A = n_A^{-1} \sum_{i=1}^{n_A} \mathbf{x}_i$ and $\boldsymbol{\mu}_B = n_B^{-1} \sum_{j=1}^{n_B} \mathbf{y}_j$. We defined the normalized centroid gap as

$$\begin{aligned} G(A, B) &= \frac{\|\boldsymbol{\mu}_A - \boldsymbol{\mu}_B\|_2}{\sqrt{(r_A^2 + r_B^2)/2}}, \\ r_A^2 = \frac{1}{n_A} \sum_{i=1}^{n_A} \|\mathbf{x}_i - \boldsymbol{\mu}_A\|_2^2, &\qquad r_B^2 = \frac{1}{n_B} \sum_{j=1}^{n_B} \|\mathbf{y}_j - \boldsymbol{\mu}_B\|_2^2. \end{aligned} \tag{4}$$

Here, r_A and r_B are the within-role root-mean-square radii, and the denominator gives equal weight to the two roles. Smaller values of G indicate less separation relative to within-role dispersion.

To assess whether Articulations retained similarity to their source Mechanisms, we compared each Articulation's source similarity with its similarities to a deterministic reference bank of 1,024 original Mechanisms, excluding its own source. Let $\mathcal{R}_i$ index the reference Mechanisms remaining after this exclusion, c_i denote the cosine similarity between Articulation i and its source Mechanism, and c_{ij} denote its similarity to reference Mechanism j. We define the source-similarity percentile as

$$R_i = \frac{1}{|\mathcal{R}_i|} \sum_{j \in \mathcal{R}_i} \left[\mathbb{I}(c_i > c_{ij}) + \frac{1}{2} \mathbb{I}(c_i = c_{ij}) \right], \tag{5}$$

where $\mathbb{I}(\cdot)$ equals 1 when the condition holds and 0 otherwise. Ties received half credit. We averaged R_i across Articulations; larger values indicate greater similarity to the source Mechanism relative to the reference Mechanisms.

4.5.3. Retrospective temporal Mechanism retrieval evaluation and blinded assessment

The retrospective evaluation covered publication years 2020–2025. For each evaluation year t, the candidate graph included only nodes and relations whose first supporting publication appeared by $t-1$. Eligible queries were Problems from year t linked to exactly one distinct realized Mechanism with complete text. Multiple Effects attached to that Mechanism were treated as part of the same solution. These criteria yielded 3,196 queries. The future Mechanisms and Effects were used for assessment only after all retrieval rankings had been fixed.

We compared nine retrieval conditions. Six first-stage systems ranked historical Mechanisms for each original Problem using TF–IDF, latent semantic analysis with 128 components, MPNet cosine similarity, one-hop graph propagation, global historical Mechanism popularity, or seeded random ranking. The graph-based system first selected the 20 historical Problems with the highest MPNet similarity, then propagated their similarity weights through pre-cutoff Problem $\xrightarrow{\text{motivates}}$ Mechanism edges. A seventh system used MPNet cosine similarity between Requirements and Articulations. Each system contributed its ten highest-ranked canonical historical Mechanisms, and their deduplicated union formed a fixed candidate pool. Two MedCPT cross-encoder conditions then reranked the entire pool using either original Problem–Mechanism pairs or Requirement–Articulation pairs. These rerankers could not add candidates to the pool.

The candidate pools contained 42–70 Mechanisms per query, with a mean of 62.5. An LLM judge based on GPT-5.5 compared each candidate with the future realized Mechanism using their complete descriptions and all attached Effects. The assessment was blinded to Requirements, Articulations, retrieval conditions, ranks, scores, paper identities, publication years, and canonical identifiers. Each pair received one of four grades: SAME CORE (3), SPECIALIZES (2; the future Mechanism retained and adapted the historical central operation), RELATED FAMILY (1) or DIFFERENT (0). Grades 2 and 3 required a shared central causal operation, acted-on object, and relevant causal intermediate; shared vocabulary, task, modality, or outcome alone was insufficient. We treated grades 2 and 3 as retrospective matches for binary ranking metrics. The assessment instruction is provided in Supplementary Note 5.

Before large-scale assessment, we compared GPT-5.5 ratings under the fixed instruction with human ratings in a blinded calibration set of 300 pairs. The sample included 50 pairs from each evaluation year and 75 from each model-assigned grade, so that agreement could be assessed across all years and all four grades. We calculated Spearman's rank correlation, exact ordinal agreement, quadratic-weighted Cohen's κ, and binary agreement using grade ≥ 2 as the positive threshold.

The complete evaluation contained 199,636 assessed pairs. For query q, let C_q denote its fixed candidate pool, with $y_{qi} = 1$ if candidate i received grade 2 or 3 and $y_{qi} = 0$ otherwise. We defined the positive and negative candidate sets as $P_q = \{i \in C_q : y_{qi} = 1\}$ and $N_q = \{i \in C_q : y_{qi} = 0\}$, respectively. Ranking metrics were evaluated on $Q = \{q : |P_q| > 0,\ |N_q| > 0\}$, the set of queries with at least one positive and one negative candidate. This resulting set contained 2,429 queries.

For retrieval method h, let $s_{qi}^{(h)}$ denote the score of candidate i for query q, with larger values ranked higher. The within-query AUROC was

$$\mathrm{AUROC}_q^{(h)} = \frac{1}{|P_q||N_q|} \sum_{i \in P_q} \sum_{j \in N_q} \left[\mathbb{I}\left(s_{qi}^{(h)} > s_{qj}^{(h)}\right) + \frac{1}{2}\mathbb{I}\left(s_{qi}^{(h)} = s_{qj}^{(h)}\right) \right]. \tag{6}$$

This score measures how often a positive candidate scores above a negative candidate for the same query, with ties receiving half credit. Query-macro AUROC assigned equal weight to each evaluated

query:

$$\mathrm{AUROC}^{(h)}_{\mathrm{macro}} = \frac{1}{|Q|} \sum_{q \in Q} \mathrm{AUROC}^{(h)}_{q} \,. \tag{7}$$

At score threshold θ, the within-query false-positive and true-positive rates were

$$\begin{aligned} \mathrm{FPR}^{(h)}_{q}(\theta) &= \frac{1}{|N_q|} \sum_{i \in N_q} \mathbb{I}\left(s^{(h)}_{qi} \geq \theta\right), \\ \mathrm{TPR}^{(h)}_{q}(\theta) &= \frac{1}{|P_q|} \sum_{i \in P_q} \mathbb{I}\left(s^{(h)}_{qi} \geq \theta\right). \end{aligned} \tag{8}$$

Varying θ produced a ROC curve for each query. Let $\widetilde{\mathrm{TPR}}^{(h)}_{q}(\upsilon)$ denote the linearly interpolated true-positive rate at a common false-positive rate $\upsilon \in [0, 1]$. The query-macro curve was $\overline{\mathrm{TPR}}^{(h)}(\upsilon) = |Q|^{-1} \sum_{q \in Q} \widetilde{\mathrm{TPR}}^{(h)}_{q}(\upsilon)$. Thus, the displayed low-FPR operating points reflected query-specific score thresholds.

Let $\mathrm{rank}^{(h)}_{q}(i)$ denote candidate i's position in the descending-score ranking, starting at 1. Shortlists followed the rankings fixed before assessment. Score ties were resolved lexicographically by canonical Mechanism identifier for first-stage retrieval and by blinded candidate identifier for MedCPT reranking. For shortlist size K, we defined

$$\mathrm{Hit@}\,K^{(h)} = \frac{1}{|Q|} \sum_{q \in Q} \mathbb{I}\left(\min_{i \in P_q} \mathrm{rank}^{(h)}_{q}(i) \leq K\right). \tag{9}$$

We reported $K = 10$, which measures the fraction of evaluated queries with at least one retrospective match among the ten highest-ranked candidates. Annual metrics used the same definitions after restricting Q to the corresponding evaluation year. Paired 95% confidence intervals were calculated from the 2.5th and 97.5th percentiles of 10,000 query-level bootstrap resamples, using the same sampled queries across methods in each resample. All ranking metrics were conditional on the fully assessed candidate pools and did not measure recall over historical Mechanisms outside those pools.

4.5.4. Human audit of RPKG construction

Human reviewers evaluated the selection of articles and the nodes and relations extracted for the three topic-specific RPKGs. To assess article selection, reviewers applied the prespecified inclusion criteria to all 595 decisions to retain or exclude an article and compared their judgments with the Retrieval Agent's decisions. To assess whether extracted relations were supported by the articles, reviewers examined three independently sampled sets: 250 candidate relations before verification, 250 relations retained after verification, and 250 discarded relations. Each relation was presented with the relevant evidence from its source article. Reviewers were not told whether the relation had passed the verification threshold.

To compare the information extracted from full articles and abstracts, we repeated extraction using only the abstracts of 135 papers for which full-text extraction results were available. For each paper, we counted complete Problem–Mechanism–Effect–Task reasoning chains and the number of functional relation types recovered under each condition. We separately assessed whether the connections within these chains were coherent. Reviewers examined 150 chains from the constructed graphs and 150 chains generated by shuffling links within individual papers. Shuffling retained the primitives extracted from each paper but changed how they were linked. The chains were presented

in randomized order. Reviewers judged whether the Mechanism could plausibly address the Problem, produce the linked Effect, and apply to the linked Task.

To assess whether the choice of LLM affected extraction, GPT-5.4, Claude Opus 4.7 and Qwen 3.7-Max independently performed the same extraction of nodes and typed relations. Human reviewers checked the extracted nodes and relations against the source articles, and we calculated node and relation accuracy from their judgments. For each pair of models, we also calculated the fraction of evaluated relations for which the models made contradictory assertions. We defined non-conflict agreement as one minus this fraction. Wilson confidence intervals were used for the reported binomial proportions.

4.5.5. Blinded human preference and model scoring of candidate hypotheses

For the human preference evaluation, we prepared three candidate hypotheses for each of 30 Problems, with ten Problems from each topic. The scHypothesizer candidate was generated by pairing the Problem with a Mechanism from the RPKG and completing the Proposer-Critic process. The Abstract-only candidate was generated from the same Problem and the original paper abstracts. The Problem-only candidate was generated from the Problem description alone. We converted each structured scHypothesizer candidate into prose and checked that it was consistent with the original structured output. We then matched the three candidates in length and writing style, removed wording that could reveal how they had been generated, and randomized their presentation order.

Six reviewers, three at doctoral level and three at master's level, each evaluated all 30 sets of candidates. For each Problem, reviewers selected the one candidate they would pursue for further method development. They could not select a tie, abstain, or reject all three candidates. This yielded 180 choices. The percentage reported for each generation condition was the number of times its candidate was selected divided by 180. The evaluation therefore measured which candidate reviewers preferred within each set; it did not establish that the selected hypothesis was scientifically valid.

We separately used Claude Opus 4.8 and GPT-5.5 to score the anonymized candidates for novelty, significance, mechanistic specificity, biological grounding, and feasibility. Each model scored every candidate three times, giving six scores per candidate for each dimension. Scores were integers from 1 to 10. The scoring instructions specified the requirements for five score bands in each dimension (Supplementary Note 3 and Supplementary Table S5). Candidate generation conditions were linked back to the scores only after scoring was complete. We summarized the scores by topic and dimension (Supplementary Table S4). The same procedure was used in the matched comparison of simple RAG candidates and MCG candidates before Proposer-Critic refinement. These scores were not used to generate candidates and were not included in the human preference analysis or the experimental evaluations. Scores for each candidate are provided in Supplementary Data 2.

4.5.6. Matched comparison of hypothesis construction

To compare how simple RAG and MCG organized retrieved evidence, we held the source corpus and evidence representation fixed while varying evidence selection and construction. For each of the 30 Problems, we generated one simple RAG candidate and one pre-critic MCG candidate in a single generation pass. The language-model backend, output schema, and Problem were held fixed. Simple RAG retrieved the five papers whose titles were most similar to the Problem and conditioned the generator on their complete novelty summaries. Pre-critic MCG instead selected novelty summaries through the typed graph pairing and organized the candidate around the matched Mechanism. The pre-critic condition omitted the subsequent multi-step Critic loop. Thus, both conditions drew on the same corpus and evidence representation but selected and organized that evidence differently.

We compared each pair with method names masked. A candidate was coded as module concatenation when it chained components from several retrieved methods without one causal operation organizing the proposal. It was coded as organized around the matched Mechanism when the graph-selected Mechanism remained the proposal's central causal operation. Auxiliary implementation steps did not change this designation when they supported the same operation. We used these categories to describe how the components of each proposal were organized. Detailed definitions, topic-level counts, and worked comparisons are provided in Supplementary Note 1.

4.5.7. Empirical assessment and score aggregation

For each executable case, we compared the prototype selected by researchers with published methods relevant to its topic. We evaluated performance on the topic's conventional task and separately tested the capabilities proposed in the hypothesis. The established methods used as baselines were specified in the benchmark plan.

For each case, we also evaluated candidates generated under the Problem-only, Abstract-only, and simple RAG conditions. In Problem-only, the same LLM backend received only the Problem description. Abstract-only additionally supplied the original article abstracts. Simple RAG supplied retrieved novelty summaries without pairing Problems and Mechanisms in the RPKG.

For each combination of dataset, task, and metric, we included the metric in score aggregation only if values could be obtained for at least two eligible methods. Metrics available for only one method were reported separately. We expressed all metrics so that higher values indicated better performance, then applied min–max scaling to $[0, 1]$ across the eligible methods for the same dataset, task, and metric [5, 20]. Evaluations for which a metric was not applicable were excluded. If a method was eligible for an evaluation but its own implementation failed to run, it received a score of zero. We did not impute other missing values or replace them with external reference scores.

For the general integration comparison, we used the scIB composite score, which assigned weights of 0.6 to biological conservation and 0.4 to batch correction [5]. For the general ambient-correction comparison and all capability-specific or secondary evaluations, we calculated the unweighted mean of the available scaled scores. The primary CONFIDE-VI alignability metric was reported separately on its original scale, as described below.

The metrics used to test each proposed capability were specified in the benchmark plan before implementation. For rare-marker-preserving ambient correction, we measured rare-cell k-nearest-neighbor (kNN) overlap with the clean-data embedding and cell-type silhouette. These metrics assessed recovery of clean-data neighborhoods and separation of cell types. We did not use retention ratios that could be maximized simply by leaving the input unchanged.

For reference mapping, the targeted board combined AUROC, AUPRC, hard-novel recall, known-cell and per-donor false-rejection rates, and accepted-known macro-F1, computed for every method that produced UNKNOWN calls natively or through a confidence threshold (Supplementary Note 2). A second targeted score assessed coverage under distribution shifts by combining conformal coverage error, coverage error calculated separately for donors or groups, Wasserstein-shift coverage slope, and mean prediction-set size, for methods that return prediction sets. For CellWDA, we combined shared-support area under the receiver-operating-characteristic curve (AUROC), the false-mixing rate for unshared cells, and the integration local inverse Simpson's index.

For CONFIDE-VI, the primary capability metric was alignability AUROC, evaluated against a chance level of 0.5. We reported this metric on its original AUROC scale without min–max scaling. We also calculated a secondary rare-cell-scaffold score as the unweighted mean of three scaled metrics:

the false-mixing rate among unalignable cells, rare-state kNN purity, and rare-state trade-off AUROC. This secondary score was reported separately and was not combined with alignability AUROC.

4.5.8. Review-style assessment of the manuscript-formatted records

Once empirical evaluations and research records were finalized, the four records underwent a review-style assessment by the same panel of six researchers who participated in the blinded candidate-selection study. The panel, comprising three doctoral and three master's-level researchers, acted as simulated reviewers under criteria adapted from the reviewer guidelines of the journal *Bioinformatics*. Each reviewer evaluated a complete record, identifying case-specific strengths and weaknesses regarding the proposed Mechanism, reported empirical evidence, and stated operational boundaries. Reviewers specifically assessed whether the record provided sufficient information to guide subsequent development decisions. Reviewers were informed that scHypothesizer had generated the records, and their evaluations were not fed back to the generation or assessment pipelines. We summarized the panel's consensus regarding record utility as a starting point for human decision-making, separate from the scientific outcomes of the benchmarks. Consequently, this evaluation focused on record interpretability and decision utility.

Data availability

The literature corpus used for RPKG Construction was retrieved from the PubMed Central Open Access Subset [43]. Articles remain subject to their respective source licenses. The literature index for the domain-scale graph was obtained from the scRNA-tools database [34]. All single-cell datasets used in the four empirical case studies are publicly available. They were obtained from Gene Expression Omnibus, ArrayExpress, CZ CELLxGENE Discover [4], Zenodo and figshare. Dataset accessions or DOIs, species, and sequencing platforms are listed in the per-method data sheets in Supplementary Note 2.

The 30 consolidated candidate method hypotheses are provided as Supplementary Data 1. Secondary blinded model ratings and anonymized human forced-choice decisions are provided as Supplementary Data 2 and 3, respectively.

Code availability

Key prompts are provided in Supplementary Note 5. The pretrained MPNet [35] and MedCPT [36] checkpoints are available as `all-mpnet-base-v2` and `MedCPT-Cross-Encoder`, respectively.

Author contributions

X.Z. designed and implemented the scHypothesizer framework, designed and conducted the experiments, prepared the figures, and wrote the original manuscript. H.G. participated in the writing and editing of the manuscript and provided guidance on figure design and presentation. L.S. provided guidance on experimental design and evaluation. X.Z. analyzed the results, and X.Z., H.G., and Z.C. interpreted and discussed the findings. Z.C. participated in the design of the framework, supervised the project, and revised the manuscript. All authors reviewed and approved the final manuscript.

Competing interests

The authors declare no competing interests.

References

[1] Sarah Aldridge and Sarah A. Teichmann. Single cell transcriptomics comes of age. *Nature Communications*, 11(1):4307, August 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-18158-5.

[2] Aviv Regev, Sarah A Teichmann, Eric S Lander, et al. Science Forum: The Human Cell Atlas. *eLife*, 6:e27041, December 2017. ISSN 2050-084X. doi: 10.7554/eLife.27041.

[3] Valentine Svensson, Roser Vento-Tormo, and Sarah A Teichmann. Exponential scaling of single-cell RNA-seq in the past decade. *Nature Protocols*, 13(4):599–604, April 2018. ISSN 1750-2799. doi: 10.1038/nprot.2017.149.

[4] CZI Cell Science Program, Shibla Abdulla, Brian Aevermann, et al. CZ CELLxGENE Discover: a single-cell data platform for scalable exploration, analysis and modeling of aggregated data. *Nucleic Acids Research*, 53(D1):D886–D900, January 2025. ISSN 1362-4962. doi: 10.1093/nar/gkae1142. URL https://doi.org/10.1093/nar/gkae1142.

[5] Malte D. Luecken, M. Büttner, K. Chaichoompu, et al. Benchmarking atlas-level data integration in single-cell genomics. *Nature Methods*, 19(1):41–50, January 2022. ISSN 1548-7105. doi: 10.1038/s41592-021-01336-8.

[6] Pierre-Luc Germain, Anthony Sonrel, and Mark D. Robinson. pipeComp, a general framework for the evaluation of computational pipelines, reveals performant single cell RNA-seq preprocessing tools. *Genome Biology*, 21(1):227, September 2020. ISSN 1474-760X. doi: 10.1186/s13059-020-02136-7.

[7] Hoa Thi Nhu Tran, Kok Siong Ang, Marion Chevrier, et al. A benchmark of batch-effect correction methods for single-cell RNA sequencing data. *Genome Biology*, 21(1):12, January 2020. ISSN 1474-760X. doi: 10.1186/s13059-019-1850-9.

[8] Mohammad Lotfollahi, Mohsen Naghipourfar, Malte D. Luecken, et al. Mapping single-cell data to reference atlases by transfer learning. *Nature Biotechnology*, 40(1):121–130, January 2022. ISSN 1546-1696. doi: 10.1038/s41587-021-01001-7. URL https://doi.org/10.1038/s41587-021-01001-7.

[9] Weijian Wang, Yihui Cen, Zezhen Lu, et al. scCDC: a computational method for gene-specific contamination detection and correction in single-cell and single-nucleus RNA-seq data. *Genome Biology*, 25(1):136, May 2024. ISSN 1474-760X. doi: 10.1186/s13059-024-03284-w. URL https://doi.org/10.1186/s13059-024-03284-w.

[10] Qingyun Wang, Doug Downey, Heng Ji, et al. SciMON: Scientific inspiration machines optimized for novelty. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 279–299, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.18. URL https://aclanthology.org/2024.acl-long.18/.

[11] Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, et al. ResearchAgent: Iterative research idea generation over scientific literature with large language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6709–6738, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.342. URL https://aclanthology.org/2025.naacl-long.342/.

[12] Daniil A. Boiko, Robert MacKnight, Ben Kline, et al. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, December 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06792-0. URL https://doi.org/10.1038/s41586-023-06792-0.

[13] Andres M. Bran, Sam Cox, Oliver Schilter, et al. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, May 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00832-8. URL https://doi.org/10.1038/s42256-024-00832-8.

[14] Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, et al. Mathematical discoveries from program search with large language models. *Nature*, 625(7995): 468–475, January 2024. ISSN 1476-4687. doi: 10.1038/s41586-023-06924-6. URL https://doi.org/10.1038/s41586-023-06924-6.

[15] Chris Lu, Cong Lu, Robert Tjarko Lange, et al. Towards end-to-end automation of AI research. *Nature*, 651(8107):914–919, March 2026. ISSN 1476-4687. doi: 10.1038/s41586-026-10265-5.

[16] Alireza Ghafarollahi and Markus J. Buehler. Sciagents: Automating scientific discovery through bioinspired multi-agent intelligent graph reasoning. *Advanced Materials*, 37(22):2413523, 2025. doi: 10.1002/adma.202413523. URL https://advanced.onlinelibrary.wiley.com/doi/abs/10.1002/adma.202413523.

[17] Yanbo Zhang, Sumeer A. Khan, Adnan Mahmud, et al. Exploring the role of large language models in the scientific method: from hypothesis to discovery. *npj Artificial Intelligence*, 1 (1):14, August 2025. ISSN 3005-1460. doi: 10.1038/s44387-025-00019-5. URL https://doi.org/10.1038/s44387-025-00019-5.

[18] Guiyao Tie, Jiawen Shi, Dingjie Song, et al. AutoResearch AI: Towards AI-powered research automation for scientific discovery. *arXiv preprint arXiv:2605.23204*, 2026. doi: 10.48550/arXiv.2605.23204. URL https://arxiv.org/abs/2605.23204.

[19] Indrajeet Mandal, Jitendra Soni, Mohd Zaki, et al. Evaluating large language model agents for automation of atomic force microscopy. *Nature Communications*, 16(1):9104, October 2025. ISSN 2041-1723. doi: 10.1038/s41467-025-64105-7. URL https://doi.org/10.1038/s41467-025-64105-7.

[20] Eser Aygün, Anastasiya Belyaeva, Gheorghe Comanici, et al. An AI system to help scientists write expert-level empirical software. *Nature*, May 2026. ISSN 1476-4687. doi: 10.1038/s41586-026-10658-6. URL https://doi.org/10.1038/s41586-026-10658-6.

[21] Florian Trost, Bide Zhang, Ines Aring, et al. An agentic framework for autonomous scientific discovery in cancer pathology. *Nature Medicine*, April 2026. ISSN 1546-170X. doi: 10.1038/s41591-026-04357-y. URL https://doi.org/10.1038/s41591-026-04357-y.

[22] Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can LLMs generate novel research ideas? a large-scale human study with 100+ NLP researchers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=M23dTGWCZy.

[23] Guangzhi Xiong, Eric Xie, Corey Williams, et al. Toward reliable scientific hypothesis generation: evaluating truthfulness and hallucination in large language models. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI '25, 2025. ISBN 978-1-956792-06-5. doi: 10.24963/ijcai.2025/873. URL https://doi.org/10.24963/ijcai.2025/873.

[24] John Dagdelen, Alexander Dunn, Sanghoon Lee, et al. Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1):1418, February 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-45563-x. URL https://doi.org/10.1038/s41467-024-45563-x.

[25] Yuxing Lu, Wei Wu, Xukai Zhao, et al. KARMA: Leveraging multi-agent LLMs for automated knowledge graph enrichment. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=k0wyi4c0Gy.

[26] Thomas Marwitz, Alexander Colsmann, Ben Breitung, et al. Predicting new research directions in materials science using large language models and concept graphs. *Nature Machine Intelligence*, 8(4):535–544, April 2026. ISSN 2522-5839. doi: 10.1038/s42256-026-01206-y. URL https://doi.org/10.1038/s42256-026-01206-y.

[27] Xuefeng Bai, Song He, Yi Li, et al. Construction of a knowledge graph for framework material enabled by large language models and its application. *npj Computational Materials*, 11(1): 51, February 2025. ISSN 2057-3960. doi: 10.1038/s41524-025-01540-6. URL https://doi.org/10.1038/s41524-025-01540-6.

[28] Halil Kilicoglu, Dongwook Shin, Marcelo Fiszman, et al. SemMedDB: a PubMed-scale repository of biomedical semantic predications. *Bioinformatics*, 28(23):3158–3160, December 2012. ISSN 1367-4803. doi: 10.1093/bioinformatics/bts591. URL https://doi.org/10.1093/bioinformatics/bts591.

[29] Daniel Scott Himmelstein, Antoine Lizee, Christine Hessler, et al. Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *eLife*, 6:e26726, September 2017. ISSN 2050-084X. doi: 10.7554/eLife.26726. URL https://doi.org/10.7554/eLife.26726.

[30] Payal Chandak, Kexin Huang, and Marinka Zitnik. Building a knowledge graph to enable precision medicine. *Scientific Data*, 10(1):67, February 2023. ISSN 2052-4463. doi: 10.1038/s41597-023-01960-3. URL https://doi.org/10.1038/s41597-023-01960-3.

[31] Fen Yan, ZhiHua Du, and Yu-An Huang. Dagformer: A graph-based domain adaptation approach for single-cell cancer drug response prediction. *PLOS Computational Biology*, 21(12):1–27, 12 2025. doi: 10.1371/journal.pcbi.1013832.

[32] Yutong Sun and Peng Qiu. Domain adaptation for supervised integration of scRNA-seq data. *Communications Biology*, 6(1):274, March 2023. ISSN 2399-3642. doi: 10.1038/s42003-023-04668-7. URL https://doi.org/10.1038/s42003-023-04668-7.

[33] Eloi Schmauch, Johannes Ojanen, Kyriakitsa Galani, et al. QClus: a droplet filtering algorithm for enhanced snRNA-seq data quality in challenging samples. *Nucleic Acids Research*, 53(1): gkae1145, January 2025. ISSN 1362-4962. doi: 10.1093/nar/gkae1145. URL https://doi.org/10.1093/nar/gkae1145.

[34] Luke Zappia, Belinda Phipson, and Alicia Oshlack. Exploring the single-cell RNA-seq analysis landscape with the scrna-tools database. *PLOS Computational Biology*, 14(6):e1006245, June 2018. doi: 10.1371/journal.pcbi.1006245. URL https://doi.org/10.1371/journal.pcbi.1006245.

[35] Kaitao Song, Xu Tan, Tao Qin, et al. MPNet: masked and permuted pre-training for language understanding. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[36] Qiao Jin, Won Kim, Qingyu Chen, et al. MedCPT: Contrastive pre-trained transformers with large-scale PubMed search logs for zero-shot biomedical information retrieval. *Bioinformatics*, 39(11):btad651, November 2023. doi: 10.1093/bioinformatics/btad651. URL https://doi.org/10.1093/bioinformatics/btad651.

[37] Lei Wang, Wanyu Xu, Yihuai Lan, et al. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada, jul 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.147. URL https://aclanthology.org/2023.acl-long.147/.

[38] Simeng Sun, Yang Liu, Shuohang Wang, et al. PEARL: Prompting large language models to plan and execute actions over long documents. In Yvette Graham and Matthew Purver, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 469–486, St. Julian's, Malta, March 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.eacl-long.29. URL https://aclanthology.org/2024.eacl-long.29/.

[39] Ehud Karpas, Omri Abend, Yonatan Belinkov, et al. MRKL systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning. *arXiv*, 2022. doi: 10.48550/arXiv.2205.00445.

[40] Shunyu Yao, Jeffrey Zhao, Dian Yu, et al. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

[41] Timo Schick, Jane Dwivedi-Yu, Roberto Dessí, et al. Toolformer: language models can teach themselves to use tools. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[42] Anthropic. Introducing the model context protocol, 2024. URL https://www.anthropic.com/news/model-context-protocol. Accessed 19 May 2026.

[43] National Library of Medicine. PMC open access subset, 2026. URL https://pmc.ncbi.nlm.nih.gov/tools/openftlist/. Accessed 19 May 2026.

[44] National Library of Medicine. PMC OA web service API, 2026. URL https://pmc.ncbi.nlm.nih.gov/tools/oa-service/. Accessed 19 May 2026.

[45] Yu A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2020. doi: 10.1109/TPAMI.2018.2889473. URL https://doi.org/10.1109/TPAMI.2018.2889473.

[46] Sampo Pyysalo, Simon Baker, Imran Ali, et al. LION LBD: a literature-based discovery system for cancer biology. *Bioinformatics*, 35(9):1553–1561, May 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty845. URL https://doi.org/10.1093/bioinformatics/bty845.

[47] Patrick Lewis, Ethan Perez, Aleksandra Piktus, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[48] Aman Madaan, Niket Tandon, Prakhar Gupta, et al. Self-refine: iterative refinement with self-feedback. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[49] Noah Shinn, Federico Cassano, Ashwin Gopinath, et al. Reflexion: language agents with verbal reinforcement learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- scHypothesizerSupplementary.pdf