# TYPED AND RECURSIVE OPERATORS FOR EVIDENCE-GROUNDED LITERATURE SYNTHESIS

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

## ABSTRACT

Producing a long literature review requires a system to find relevant papers, organize their evidence, write individual sections, and combine those sections without losing the link between claims and sources. We present SLRGP, a workflow that divides this process into nine operators. Each operator has a defined input, output, and responsibility, so its implementation can be replaced and evaluated independently.

We test the design in four ways. First, we recover section trees from 2,221 scholarly reviews and show that their organization can be distinguished from closely matched artificial alternatives. Second, citation data trains an effective paper ranker, and section data supports recognition of common organization patterns. Third, controlled replacements show that evidence-aware organization and validation improve 1,000-word reviews. Fourth, when every method receives the same 32 evidence cards, hierarchical writing and integration outperform naive chunking, a fixed outline, and flat generation on all six long-form topics. These results establish explicit operator interfaces as a common basis for controlled experimentation and panel-preferred long-form synthesis.

## 1 INTRODUCTION

A literature-review system must make several decisions before it can produce useful prose. It must turn a research question into searches, select admissible evidence, rank the candidate papers, and choose an organization. During writing, it must also detect invalid intermediate results and preserve the link between each claim and its source. Long reviews add a further challenge: sections written from different evidence subsets must form one coherent argument.

Many existing systems perform these steps inside a single prompt or tightly coupled pipeline. This design makes it difficult to identify which step caused an improvement or an error. It also makes controlled comparison difficult, because changing one component may change the rest of the workflow.

We address this problem by representing literature synthesis as a sequence of typed operators. Here, an operator is a component with a defined input and output. Its contract states which parts of the shared record it may change, which links it must preserve, and what happens when a check fails. Different implementations—including rules, retrieval models, supervised predictors, and language models—can therefore be compared through the same interface.

We implement this design in the Systematic Literature Review Generative Pipeline (SLRGP). Its nine operators expand the query, locate papers, filter ineligible sources, rank candidates, organize the evidence, validate the current state, prepare source notes, write local sections, and combine them. For a long review, the organizer divides the topic into smaller subproblems. The system writes the smallest sections from local evidence and then adds higher-level synthesis as it combines them.

The paper makes four contributions:

1. It defines a shared record that tracks sources, structure, generated text, and budgets, together with nine operator contracts that govern how this record may change.
2. It constructs a structural corpus from 2,221 scholarly reviews and uses their citations and section trees to identify decisions that can be learned from existing reviews.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

3. It changes one operator at a time to measure how evidence-aware organization and validation affect short reviews.
4. It separates hierarchical writing from simple chunking and compares them using the same source material in long-form generation.

Together, these contributions provide a common basis for learning, replacing, inspecting, and testing the main synthesis decisions independently. The nine-operator design also allows one part of the workflow to be extended without redesigning the complete system.

## 2 Related Work

**Review automation and scientific synthesis.** Systematic-review methodology separates scope definition, search, selection, synthesis, and reporting (Page et al., 2021; Higgins et al., 2024). Earlier automation work treated review construction as a collection of machine-learning tasks (Tsafnat et al., 2014; Marshall & Wallace, 2019; van de Schoot et al., 2021). Recent systems combine retrieval, outline construction, writing, and citation handling (Wang et al., 2024; Yan et al., 2025; Bao et al., 2025; Liang et al., 2025; Hu et al., 2024; Asai et al., 2026). Hierarchical catalogues and outline-guided generation further show that document structure matters for long-form writing (Zhu et al., 2023; Shao et al., 2024). SLRGP builds on this work by defining explicit interfaces between these steps and testing alternative implementations at the same interface.

**Modular workflows and typed composition.** Workflow patterns and interface-based composition separate what a component must do from how it does it (van der Aalst et al., 2003; de Alfaro & Henzinger, 2001). We apply this principle to literature synthesis. Each contract limits the changes an operator may make, preserves source tracking, and allows different implementations to be compared under the same surrounding conditions.

**Grounded generation and evaluation.** A citation is useful only when the cited source supports the associated claim (Gao et al., 2023; Min et al., 2023). Model-based evaluation can score many long outputs, but its judgments may depend on position, verbosity, or model family (Zheng et al., 2023; Panickssery et al., 2024). We therefore combine automatic checks, randomized pairwise comparisons by several model families, and a separate claim–source support test. The appendix reports how this evaluation responds to controlled defects (Verga et al., 2024).

## 3 Typed Operator Framework

### 3.1 A shared record for the review

The system stores its current work in one versioned record:

$$s = \langle q, D, \Gamma, \kappa, x, \pi, b \rangle \in \mathcal{S}, \tag{1}$$

where $q$ is the question, $D$ is the collected evidence, $\Gamma$ is the ordered section tree, $\kappa$ stores validation results, and $x$ is the text written so far. The provenance ledger $\pi$ connects source records, evidence notes, claims, and citations. The budget record $b$ limits retrieval, context, retries, recursion depth, generation, and output length. Keeping these items in one record lets the system trace each output and stop predictably when a budget is exhausted. The fields are versioned collections: $q$ includes the original question and its expanded queries, $D$ records candidate, eligible, and ranked views of the evidence, and $x$ contains both local and assembled text.

The interfaces distinguish objects by their intended use. Search starts with a review question $\mathcal{Q}$, produces expanded queries $\mathcal{Q}^+$, retrieves candidates $\mathcal{D}_c$, filters them to eligible evidence $\mathcal{D}_e$, and returns a ranked list $\Pi(\mathcal{D}_e)$. Organization turns that list into an ordered tree $\mathcal{G}$ and then a validated tree $\mathcal{G}_v$. Writing uses evidence cards $\mathcal{A}$ to produce local sections $\mathcal{X}_\ell$ and a complete document $\mathcal{X}_g$.

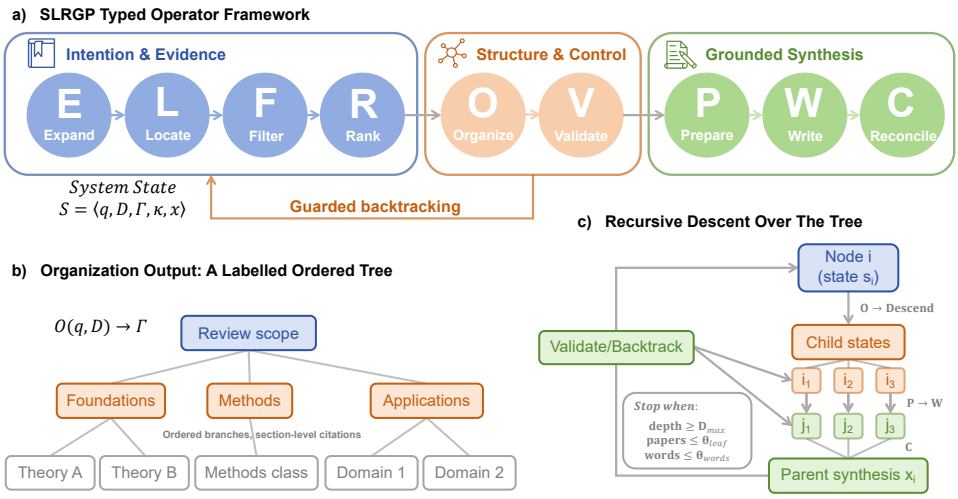


Figure 1: The SLRGP workflow. The first four operators find, filter, and rank papers. Organization and validation build and check the section structure. The final three operators prepare source notes, write local sections, and combine them. Long reviews are assembled from the smallest sections upward.

### 3.2 Operator Contracts

The nine interfaces are

$$
\begin{aligned}
&E : \mathcal{Q} \times \mathcal{B} \to \mathcal{Q}^+, && L : \mathcal{Q}^+ \times \mathcal{I} \times \mathcal{B} \to \mathcal{D}_c,\\
&F : \mathcal{Q} \times \mathcal{D}_c \times \mathcal{K} \to \mathcal{D}_e, && R : \mathcal{Q} \times \mathcal{D}_e \to \Pi(\mathcal{D}_e),\\
&O : \mathcal{Q} \times \Pi(\mathcal{D}_e) \times \mathcal{B} \to \mathcal{G}, && V : \mathcal{S} \times \mathcal{B} \to \mathcal{S} \times \{\text{accept}, \text{retry}, \text{stop}\},\\
&P : \Pi(\mathcal{D}_e) \times \mathcal{G}_v \to \mathcal{A}, && W : \mathcal{Q} \times \mathcal{A} \times \mathcal{B} \to \mathcal{X}_\ell,\\
&C : \mathcal{G}_v \times \mathcal{X}_\ell^* \times \mathcal{B} \to \mathcal{X}_g.
\end{aligned}
$$

The symbols follow the workflow: $E$ expands the query, $L$ locates papers, $F$ filters them, and $R$ ranks them. $O$ organizes the evidence and $V$ checks the resulting state. Finally, $P$ prepares evidence cards, $W$ writes local sections, and $C$ combines those sections.

For each operator $i$, its contract specifies a precondition $\mathrm{Pre}_i(s)$, a permitted write set $\mathrm{Write}_i$, and a postcondition $\mathrm{Post}_i(s, s')$. Three invariants hold after every accepted transition: every evidence card resolves to a source in $D$, every citation in $x$ resolves through $\pi$, and every active writing target belongs to the validated tree $\Gamma_v$. An accepted transition therefore has the form

$$
s \xrightarrow{i} s' \quad \Longleftrightarrow \quad \mathrm{Pre}_i(s) \wedge \mathrm{Post}_i(s, s') \wedge \mathrm{Inv}(s'), \tag{2}
$$

with fields outside $\mathrm{Write}_i$ unchanged. Table 1 summarizes the responsibility assigned to each interface.

Each contract lists the fields an operator may change. A successful call must return a valid object while preserving source identifiers and claim–citation links. These rules make the execution trace auditable and create clear points for checking source validity and claim support.

The validator separates required conditions from quality preferences. Required conditions include valid source identifiers, non-empty branches, traceable evidence cards, and well-formed citation links. Quality checks cover branch balance and evidence sufficiency. When a check fails, the system follows a registered retry rule or stops after using its retry budget.

Table 1: Operator contracts. Each interface owns one decision while preserving source identity, provenance, and validated structure as applicable.

| Operator | Reads | Produces or updates | Preserves |
|---|---|---|---|
| $E$ Expand | $q, b$ | expanded queries | source namespace |
| $L$ Locate | $q^+, \mathcal{I}, b$ | candidate evidence | query trace |
| $F$ Filter | $q, D_c, \mathcal{K}$ | eligible evidence | source identifiers |
| $R$ Rank | $q, D_e$ | ordered evidence | evidence membership |
| $O$ Organize | $q, \Pi(D_e), b$ | ordered section tree | source assignments |
| $V$ Validate | $s, b$ | checks and action | accepted state fields |
| $P$ Prepare | $\Pi(D_e), \Gamma_v$ | evidence cards | source links |
| $W$ Write | $q, \mathcal{A}, b$ | local sections | claim–source links |
| $C$ Combine | $\Gamma_v, \mathcal{X}^*_\ell, b$ | complete document | order and citations |

### 3.3 Recursive execution

For a long review, the organizer divides an internal node $v$ into child problems $\{s_{v_1}, \ldots, s_{v_k}\}$. Each child receives a narrower question, its own evidence, and a word budget. At the bottom of the tree, the writer produces a local section, $x_{v_i} = W(q_{v_i}, A_{v_i}, b_{v_i})$. The parent then writes a synthesis bridge and combines the completed child sections in their planned order:

$$x_v = C(\Gamma_v, [x_{v_1}, \ldots, x_{v_k}], b_v). \tag{3}$$

This process preserves completed local sections and adds integration where branches meet. It therefore maintains local source links while improving the argument across sections. Recursion ends when the depth budget reaches zero, the evidence already fits in one leaf, or the remaining word budget is too small for a useful split.

### 3.4 What is learned

Each interface can learn from a different kind of trace. In general, the training set for interface $i$ is $\mathcal{D}_i = \{(u_n, y_n)\}_{n=1}^{N_i}$, where $u_n$ is the information available to that interface and $y_n$ is an observed decision. Citations in published reviews provide labels for paper ranking. Existing section trees provide labels for the principle used to divide and order sections.

The current organization learner identifies the pattern used by an observed section tree. Its inputs include features from that tree, and its outputs are the division principle and ordering relation. This result shows that published reviews contain learnable organization patterns. In the present system, this schema-recognition model characterizes observed structures, while recursive generation instantiates the same $O$ interface with a language-model organizer. The shared contract keeps these two roles explicit and comparable.

## 4 Evidence and Evaluation Design

### 4.1 Structural corpus and splits

We collected 2,862 English-language arXiv reviews from eleven disciplines and parsed their available LaTeX sources. The parser recovered usable section trees for 2,223 reviews. After removing two reviews with fewer than two top-level sections, the corpus contained 2,221 reviews and 14,332 section nodes. For nodes with a defined value, an average of 93.5% of citations could be linked to a child section. We split complete reviews across training, validation, and test sets so that sections from the same review never appeared in different sets. Organization-pattern recognition used an 80/20 split, and ranking used a 70/15/15 split. OpenAlex supplied bibliographic metadata and record links (Priem et al., 2022).

### 4.2 Implementations and controlled comparisons

The retrieval operator combined BM25 and dense retrieval with reciprocal-rank fusion (Robertson & Zaragoza, 2009; Cormack et al., 2009). The ranker used a LambdaMART-style gradient-boosted

model trained from papers cited in existing reviews (Burges, 2010; Ke et al., 2017; Järvelin & Kekäläinen, 2002). A logistic-regression model with TF–IDF features recognized organization patterns. We fixed all language-model versions and settings before running the generation experiments.

To measure one operator at a time, the short-review experiment kept the topic, source bundle, writing model, source-tracking method, and target length constant. Only the selected operator changed. In the long-review experiment, all four methods received the same evidence pool and the same 32 source notes assigned to the smallest sections. Statistical tests treated each review or topic as one independent unit. We used paired bootstrap intervals, exact tests, and Benjamini–Hochberg correction for the small topic-level comparisons (Benjamini & Hochberg, 1995).

### 4.3 Evaluation protocol

We formed an evaluation panel from three different model families. The panel received anonymized outputs in random order and scored organization, critical synthesis, citation plausibility, and overall coherence. We calibrated the panel by deliberately reordering sections, altering citations, adding repetition, and removing synthesis from published reviews. We also checked whether individual claims were supported by their cited sources. The appendix reports the full calibration results and agreement between evaluators.

## 5 Results

### 5.1 Recovered trees retain a reliable organizational signal

We paired 2,881 real section structures with artificial alternatives that matched their discipline, tree depth, title length, number of branches, number of citation-bearing children, and topic neighborhood. Evaluators selected the real structure in 56.5% of pairs, with ROC–AUC 0.581. The mean score advantage for the real structure was 0.082 (95% bootstrap CI $[0.067, 0.096]$; permutation $p < 10^{-4}$).

Because the artificial alternatives already match the main surface properties, this reliable difference reflects relationships among sibling sections. The result confirms that the recovered trees preserve meaningful organization patterns for the later learning experiments.

### 5.2 Learning is most effective for ranking and organization

The benefit of learning differed by step. The learned query-expansion model changed the share of linked papers found in the top 300 results from 0.2996 to 0.2982 and reduced later ranking quality. We therefore kept the original query policy. The learned eligibility model achieved AUC 0.707, showing that it captured semantic information, while the simple threshold performed better at the required high-recall operating point. For paper location, combining sparse and dense retrieval improved the candidate set. Their union covered 53.4% of linked papers at depth 3,000, making broader candidate retrieval the clearest next source of improvement.

Ranking produced the largest learning gain. Compared with reciprocal-rank fusion, the learned ranker increased nDCG@10 from 0.259 to 0.412. The difference was 0.153 (95% CI $[0.113, 0.201]$, $p \approx 3 \times 10^{-10}$). It also increased Recall@50 from 0.421 to 0.594 and MRR from 0.468 to 0.674.

Published section trees also revealed learnable organization patterns. The model identified the principle used to divide a topic with accuracy 0.518, compared with 0.395 for the strongest baseline. At the review level, the mean gain was 0.110 (95% CI $[0.078, 0.141]$). It identified the relation used to order sections with accuracy 0.431 versus 0.388, a review-level gain of 0.038 (95% CI $[0.007, 0.069]$). These results provide a supervised foundation for representing organization decisions at the $O$ interface.

### 5.3 Changing one component reveals the effects of organization and validation

We first tested whether organizing papers by their content improves a short review. The control simply grouped consecutive papers in retrieval order. On eight 1,000-word topics, evidence-aware

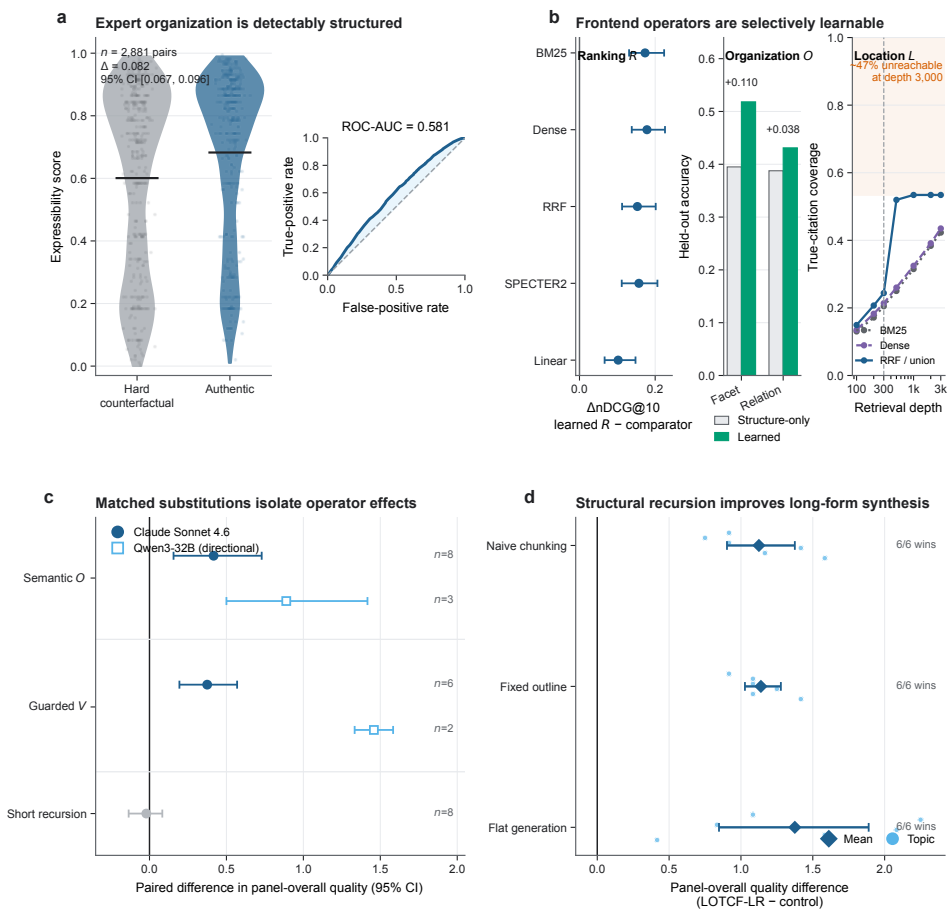


Figure 2: Four main results. Real review structures differ reliably from closely matched artificial alternatives; citations and section trees support learning; evidence-aware organization and validation improve short reviews; and hierarchical writing improves long reviews when every method receives the same evidence.

organization increased overall quality by 0.417 (95% CI $[0.156, 0.729]$; exact $p = 0.016$; FDR-adjusted $q = 0.047$), with seven wins and one loss. The largest gains appeared in organization ($+0.917$) and global coherence ($+0.708$).

We then tested validation under deliberately introduced stress. The guarded version checks intermediate results and retries when a required condition fails; the control continues without this check. Across six eligible pairs, validation increased overall quality by 0.375 (95% CI $[0.194, 0.569]$; exact $p = 0.031$; $q = 0.047$), winning on all six topics. Two unguarded outputs fell below the length threshold fixed before evaluation. A companion experiment with an open-weight model preserved both positive directions among its eligible pairs. Citation plausibility remained stable while organization and validation improved.

### 5.4 Structural recursion improves long-form panel scores

We use *structural recursion* to mean two operations: the system reorganizes evidence at internal nodes and writes a synthesis passage when it combines child sections. An early chunking condition performed neither operation; it only divided fixed groups into adjacent writing units. The complete configuration used in the main comparison adds both operations, distinguishing the evaluated structural-recursion design from simple segmentation.

We compared the complete method with naive chunking, a fixed outline, and flat generation on six long-form topics. Every method used the same evidence pool and the same 32 evidence cards for

the smallest sections. Structural recursion increased overall quality by 1.125 over naive chunking (95% CI $[0.903, 1.375]$), 1.139 over a fixed outline ($[1.028, 1.278]$), and 1.375 over flat generation ($[0.847, 1.889]$). It won on all six topics in every comparison (exact $p = 0.031$; FDR-adjusted $q = 0.039$). Organization, global coherence, and critical synthesis improved throughout, while citation plausibility remained comparable to the strongest structured controls.

Using identical evidence cards keeps source selection fixed and focuses the comparison on how the methods organize and combine the material. The reported contrast evaluates the complete structural-recursion configuration, in which internal reorganization and parent-level integration operate together.

### 5.5 Native systems show a favorable deployment profile

We also ran the native workflows of SLRGP, AutoSurvey, SURVEYFORGE, and SurveyGen on eight topics with the same generation backbone and a 2023 publication cutoff. Their mean output lengths were 4,032, 60,483, 12,128, and 5,603 words, respectively. Among sampled claims, the proportions supported by their cited sources were 0.835, 0.535, 0.559, and 0.707. Mean generation costs were \$0.65, \$11.38, \$7.46, and \$3.01 per output.

For quality comparison, evaluators read matched 4,000-word excerpts. They preferred SLRGP to AutoSurvey and SURVEYFORGE. SurveyGen provided the closest comparison, with no resolved difference after correction. The combined results show that SLRGP occupies a favorable region of the quality–support–cost trade-off. This native comparison describes each workflow in its intended deployment configuration; the controlled replacement experiments provide the matched-input evidence for individual design choices.

## 6 Discussion

The experiments show that different stages benefit from different methods. Citation data is highly effective for learning to rank papers. The retrieval results show where a larger candidate pool can improve that ranker. Section trees reveal organization patterns that characterize decisions at the organizer interface. Controlled replacements show separate gains from evidence-aware grouping and validation. Most clearly, the long-form study shows that the complete hierarchical configuration—which reorganizes evidence and connects completed sections—outperforms the three tested alternatives.

Explicit interfaces make these findings possible. They state where evidence may change, where document structure is created, where checks occur, and which information must survive a component replacement. The same design also makes the system easier to inspect: source links remain visible, errors can be tied to a particular stage, and failed checks follow a recorded recovery rule. The contract table makes these responsibilities explicit across all nine interfaces.

**Evidence scope.** The short-form experiments change one operator at a time, and the long-form experiment fixes the source material across all methods. These controls support a consistent result: evidence-aware organization and guarded validation improve short reviews, while the complete hierarchical configuration improves panel-rated long-form quality. The panel calibration establishes sensitivity to the structural and citation-validity perturbations used in the study, and the separate claim–source check complements those document-level scores.

## 7 Conclusion

SLRGP divides literature synthesis into nine components with explicit inputs, outputs, and responsibilities. This design allows each decision to be learned, replaced, and tested while the rest of the workflow stays fixed. Experiments show that citation data trains an effective ranker, published section trees contain learnable organization patterns, and organization and validation improve short reviews. On long reviews, hierarchical organization and integration outperform three alternative writing strategies when every method receives the same evidence. These results establish explicit operators as a practical foundation for evidence-grounded literature synthesis.

### Reproducibility Statement

The submission package contains source-tracking records, review-level data splits, learning code, generation traces, prompts, evaluation logs, and scripts for the reported analyses. The appendix records exclusions, model roles, topic counts, budgets, protocol changes, and evaluation calibration results.

### Ethics Statement

The study uses publicly available scholarly documents and bibliographic metadata and does not involve private personal data. Generated reviews are research outputs rather than substitutes for expert evidence assessment. Source identifiers and provenance records are retained so that cited claims can be checked against the underlying literature.

### AI Use Statement

Language models were used in two roles within the research: as components of the studied synthesis workflow and as automated evaluators. AI-based coding and writing assistants also supported experimental implementation, English translation, editorial restructuring, readability improvements, literature discovery, and the presentation of the operator framework. The authors defined and checked the research questions, experimental designs, analyses, citations, and claims, reviewed all AI-assisted work, and take responsibility for the final content.

## References

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'Arcy, David Wadden, Matt Latzke, Minyang Tian, Pan Ji, Shengyan Liu, Hao Tong, Bohao Wu, Yanyu Xiong, Luke Zettlemoyer, Graham Neubig, Dan Weld, Doug Downey, Wen-tau Yih, Pang Wei Koh, and Hannaneh Hajishirzi. Synthesizing scientific literature with retrieval-augmented language models. *Nature*, 650:857–863, 2026. doi: 10.1038/s41586-025-10072-4.

Tong Bao, Mir Tafseer Nayeem, Davood Rafiei, and Chengzhi Zhang. SurveyGen: Quality-aware scientific survey generation with large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2712–2736, 2025.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995.

Christopher J. C. Burges. From RankNet to LambdaRank to LambdaMART: An overview. Technical Report MSR-TR-2010-82, Microsoft Research, 2010.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 758–759, 2009.

Luca de Alfaro and Thomas A. Henzinger. Interface automata. In *Proceedings of the 8th European Software Engineering Conference Held Jointly with the 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, pp. 109–120, 2001. doi: 10.1145/503209.503226.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6465–6488, 2023. doi: 10.18653/v1/2023.emnlp-main.398.

Julian P. T. Higgins, James Thomas, Jacqueline Chandler, Miranda Cumpston, Tianjing Li, Matthew J. Page, and Vivian A. Welch (eds.). *Cochrane Handbook for Systematic Reviews of Interventions*. Cochrane, version 6.5 edition, 2024. URL `https://training.cochrane.org/handbook/current`.

Yuntong Hu, Zhuofeng Li, Zheng Zhang, Chen Ling, Raasikh Kanjiani, Boxin Zhao, and Liang Zhao. HiReview: Hierarchical taxonomy-driven automatic literature review generation. *Preprint at https://arxiv.org/abs/2410.03761*, 2024.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pp. 3146–3154, 2017.

Xun Liang, Jiawei Yang, Yezhaohui Wang, Chen Tang, Zifan Zheng, Shichao Song, Simin Niu, Hanyu Wang, Bo Tang, Feiyu Xiong, Keming Mao, and Zhiyu Li. SurveyX: Academic survey automation via large language models. *Preprint at https://arxiv.org/abs/2502.14776*, 2025.

Iain J. Marshall and Byron C. Wallace. Toward systematic review automation: A practical guide to using machine learning tools in research synthesis. *Systematic Reviews*, 8:163, 2019. doi: 10.1186/s13643-019-1074-9.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12076–12100, 2023. doi: 10.18653/v1/2023.emnlp-main.741.

Matthew J. Page, Joanne E. McKenzie, Patrick M. Bossuyt, Isabelle Boutron, Tammy C. Hoffmann, Cynthia D. Mulrow, Larissa Shamseer, Jennifer M. Tetzlaff, Elie A. Akl, Sue E. Brennan, Roger Chou, Julie Glanville, Jeremy M. Grimshaw, Asbjørn Hróbjartsson, Manoj M. Lalu, Tianjing Li, Elizabeth W. Loder, Evan Mayo-Wilson, Steve McDonald, Luke A. McGuinness, Lesley A. Stewart, James Thomas, Andrea C. Tricco, Vivian A. Welch, Penny Whiting, and David Moher. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, 372: n71, 2021. doi: 10.1136/bmj.n71.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM evaluators recognize and favor their own generations. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024.

Jason Priem, Heather Piwowar, and Richard Orr. OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. *Preprint at https://arxiv.org/abs/2205.01833*, 2022.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

Yijia Shao, Yucheng Jiang, Theodore Kanell, Peter Xu, Omar Khattab, and Monica Lam. Assisting in writing Wikipedia-like articles from scratch with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6252–6278, 2024. doi: 10.18653/v1/2024.naacl-long.347.

Guy Tsafnat, Paul Glasziou, Miew Keen Choong, Adam Dunn, Filippo Galgani, and Enrico Coiera. Systematic review automation technologies. *Systematic Reviews*, 3:74, 2014. doi: 10.1186/2046-4053-3-74.

Rens van de Schoot, Jonathan de Bruin, Raoul Schram, Parisa Zahedi, Jan de Boer, Felix Weijdema, Bianca Kramer, Martijn Huijts, Maarten Hoogerwerf, Gerbrich Ferdinands, Albert Harkema, Joukje Willemsen, Yongchao Ma, Qixiang Fang, Sybren Hindriks, Lars Tummers, Daniel L. Oberski, et al. An open source machine learning framework for efficient and transparent systematic reviews. *Nature Machine Intelligence*, 3:125–133, 2021. doi: 10.1038/s42256-020-00287-7.

Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, Bartek Kiepuszewski, and Alistair P. Barros. Workflow patterns. *Distributed and Parallel Databases*, 14(1):5–51, 2003. doi: 10.1023/A:1022883727209.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. Replacing judges with juries: Evaluating LLM generations with a panel of diverse models. *Preprint at* `https://arxiv.org/abs/2404.18796`, 2024.

Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Min Zhang, Qingsong Wen, Wei Ye, Shikun Zhang, and Yue Zhang. AutoSurvey: Large language models can automatically write surveys. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024.

Xiangchao Yan, Shiyang Feng, Jiakang Yuan, Renqiu Xia, Bin Wang, Lei Bai, and Bo Zhang. SurveyForge: On the outline heuristics, memory-driven generation, and multi-dimensional evaluation for automated survey writing. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023.

Kun Zhu, Xiaocheng Feng, Xiachong Feng, Yingsheng Wu, and Bing Qin. Hierarchical catalogue generation for literature review: A benchmark. *Preprint at* `https://arxiv.org/abs/2304.03512`, 2023.

# A Experimental Overview

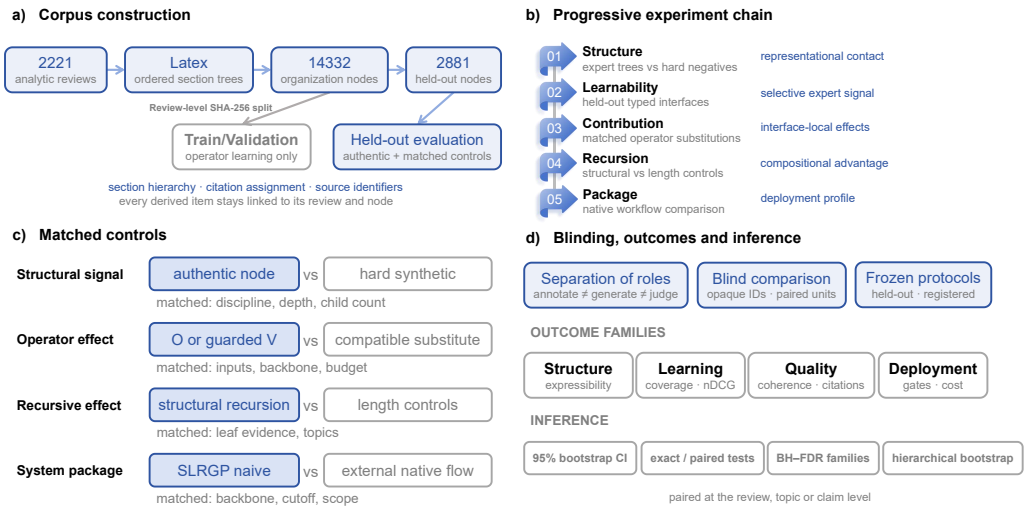


Figure 3: How the corpus was built, how each experiment extends the previous one, and how methods were compared without revealing their identities.

The experiments proceed through four linked evaluations. The representation test measures whether the recovered corpus preserves meaningful section structure. Interface prediction identifies decisions that can be learned from citations and section trees. Controlled replacements isolate changes at one operator, and the long-form comparison evaluates hierarchical organization under a shared evidence pool. A final native comparison describes complete systems in their intended deployment configurations.

## B Corpus Construction and Representation Test

We parsed 2,862 candidate reviews with deterministic rules. The parser expanded `\input` and `\include` files, removed comments, identified the main file, recovered section commands, and assigned citations to sections. It returned 2,223 usable trees. We removed two reviews with fewer than two top-level sections, leaving 2,221 reviews.

Before adding a common root for each review, the trees contained 12,111 branching nodes. The final corpus contained 14,332 nodes: 2,221 at depth 0, 7,834 at depth 1, and 4,277 at depth 2.

For each of 2,881 held-out nodes, we created two artificial alternatives. Tier 1 matched the real node by discipline and number of branches. Tier 2 also matched topic neighborhood, tree depth, title length, number of citation-bearing children, and local plausibility; this stricter version was the primary control. An early annotation pass revealed that visible identifiers encoded the condition. We corrected the protocol by using opaque identifiers and random order, then regenerated the final results in Table 2.

Table 2: Real versus matched artificial organization structures.

| Control | Pairwise acc. | ROC–AUC | Mean $\Delta$ | 95% CI |
|---|---|---|---|---|
| Tier 1 matched | 0.592 | 0.594 | +0.105 | [0.090, 0.120] |
| Tier 2 hard matched | 0.565 | 0.581 | +0.082 | [0.067, 0.096] |

The eleven disciplines included economics and quantitative finance. Their combined held-out Tier-2 slice contained 132 pairs with mean difference $+0.077$ (95% CI $[+0.005, +0.149]$; permutation $p = 0.037$). This analysis was fixed before examining the discipline-level result and confirms the organization signal in economics and quantitative finance.

## C Interface Prediction Details

**Expansion.** A logistic model selected ten additional search terms from the review title, abstract, discipline, and top-50 BM25 results. On held-out reviews, Coverage@300 changed from 0.2996 to 0.2982 and downstream nDCG@10 changed from 0.412 to 0.305. We therefore used the stronger original query policy in later experiments.

**Location.** We evaluated sparse and dense retrieval to depth 3,000 on 1,102 reviews. Together, the two methods found 53.4% of linked papers. Reciprocal-rank fusion (RRF) placed 24.4% in the top 300. A learned fusion model increased this value by 0.0113, and adding a citation-supervised bi-encoder increased it by 0.018. The remaining difference between the top-300 result and the union shows the opportunity for better candidate selection.

**Filtering.** Three models independently labelled stratified samples from the top 500 candidates, and their consensus provided eligibility labels. The learned filter reached AUC 0.707. At the high-recall point chosen before evaluation, it obtained recall 0.926 and precision 0.761; a simple RRF threshold obtained 0.954 and 0.766. We kept the stronger threshold in the main pipeline, while the AUC result shows that eligibility contains learnable semantic signal.

**Ranking.** Papers cited in the body of existing reviews served as positive ranking examples. An initial audit found that 91.0% of these links were correct. Adding surname and year checks for short, generic titles removed 185 same-title errors and raised estimated precision to 96.2% (95% CI $[0.931, 0.987]$). We then simulated label noise at one, two, and three times the remaining error rate. The learned ranker outperformed RRF in all 200 draws at every noise level.

**Organization-schema recognition.** Each section node received two labels. The first described how its children divide the topic, and the second described how those children are ordered. Annotator agreement was 0.718 ($\kappa = 0.633$) for the division label and 0.627 ($\kappa = 0.488$) for the ordering label. The classifier used the current, parent, and child titles together with depth, review length, discipline,

Table 3: Held-out ranking at the top-300 interface ($n = 153$ reviews).

| Method | nDCG@10 | Recall@50 | MRR |
|---|---|---|---|
| RRF input order | 0.259 | 0.421 | 0.468 |
| SPECTER2 cosine | 0.255 | 0.471 | 0.467 |
| Linear pairwise | 0.310 | 0.460 | 0.553 |
| Learned LambdaMART ranker | **0.412** | **0.594** | **0.674** |

and number of children. The experiment shows that these organization patterns can be recognized and provides supervised labels for organization decisions.

## D One-Component Replacement Details

The first pilot generated 40 outputs across five conditions. It showed that output lengths were comparable on only one of eight topics and that one intended ranking change also altered the top-level grouping. We used this audit to design a stricter main experiment.

In the main experiment, every condition used the same root evidence, candidate order, number and size of groups, writing model, and 1,000-word target. All 40 outputs completed, 38 met the design criteria, and the blinded panel returned 120 valid scores. The validation comparison contained six pairs because two outputs without guards fell below the 850-word threshold fixed in advance.

Table 4: One-component replacements; positive differences favor the complete operator.

| Contrast | $n$ | $\Delta$ | 95% CI | $p/q$ | W/T/L |
|---|---|---|---|---|---|
| Semantic $O$ vs. rank slabs | 8 | +0.417 | [0.156, 0.729] | .016/.047 | 7/0/1 |
| Short recursion vs. flat | 8 | $-0.021$ | $[-0.135, 0.083]$ | .875/.875 | 4/1/3 |
| Guarded $V$ vs. unguarded | 6 | +0.375 | [0.194, 0.569] | .031/.047 | 6/0/0 |

As a robustness check, Qwen3-32B generated 16 outputs across four topics. Thirteen met the main length criterion, producing three organization pairs and two validation pairs. Every eligible pair favored the complete operator. The same direction held when all outputs were included.

## E Structural Recursion Details

Stage A covered six topics, three target lengths (1,200, 3,600, and 9,000 words), and three writing strategies. All 54 outputs met the design criteria. An implementation audit showed that the condition initially called recursive used fixed groups, wrote them as adjacent units, and concatenated the results. It therefore measured chunking. The result established a useful boundary and led to the complete structural design in Stage B.

Stage B added organization at internal nodes and integration when completed sections were combined. A one-topic diagnostic recorded six organization calls, 28 downward steps, 11 bounded section writes, and seven upward integration steps. The main run covered six long topics. It gave the complete method and all three controls the same evidence pool and 32 evidence cards. Retrieval and ranking remained fixed at the root so that the experiment focused on organization, writing, and integration.

Table 5: Stage-B structural-recursion comparisons over six long-form topics.

| Control | Overall $\Delta$ | 95% CI | $p/q$ | W/T/L |
|---|---|---|---|---|
| Naive chunking | +1.125 | [0.903, 1.375] | .031/.039 | 6/0/0 |
| Fixed outline | +1.139 | [1.028, 1.278] | .031/.039 | 6/0/0 |
| Flat generation | +1.375 | [0.847, 1.889] | .031/.039 | 6/0/0 |

Organization, global coherence, and critical synthesis improved in every comparison. Citation plausibility matched naive chunking and the fixed outline and was higher in direction than flat generation ($\Delta = 1.000$, $q = 0.072$).

## F Native Workflow Comparison

The experiment began with 12 topics in four categories. Before quality evaluation, we selected the first two topics in each category to keep cost within the recorded budget. This produced eight topics and 32 completed system runs. None cited work published after the 2023 cutoff. Every citation identifier resolved for SLRGP, AutoSurvey, and SURVEYFORGE; the resolution rate was 0.921 for SurveyGen. All eight normalized SURVEYFORGE outputs contained detectable prompt or meta text.

Table 6: Native deployment results under each workflow's default settings.

| System | Mean words | Sampled support | Cost/output |
|---|---|---|---|
| SLRGP | 4,032 | 0.835 | $0.65 |
| AutoSurvey | 60,483 | 0.535 | $11.38 |
| SURVEYFORGE | 12,128 | 0.559 | $7.46 |
| SurveyGen | 5,603 | 0.707 | $3.01 |

We sampled 12 cited claims from each output, producing 384 claims. After fixing a pagination error in metadata retrieval, we rebuilt every source packet and repeated the judgments. Sources were accessible for 94/96 SLRGP claims, 86/96 AutoSurvey claims, 94/96 SURVEYFORGE claims, and 92/96 SurveyGen claims. We also measured how often each output included the core papers cited by the source review. The values were 0.045, 0.045, 0.055, and 0.100, respectively. This coverage measure complements the claim-support score and makes the support–coverage profile visible for every system.

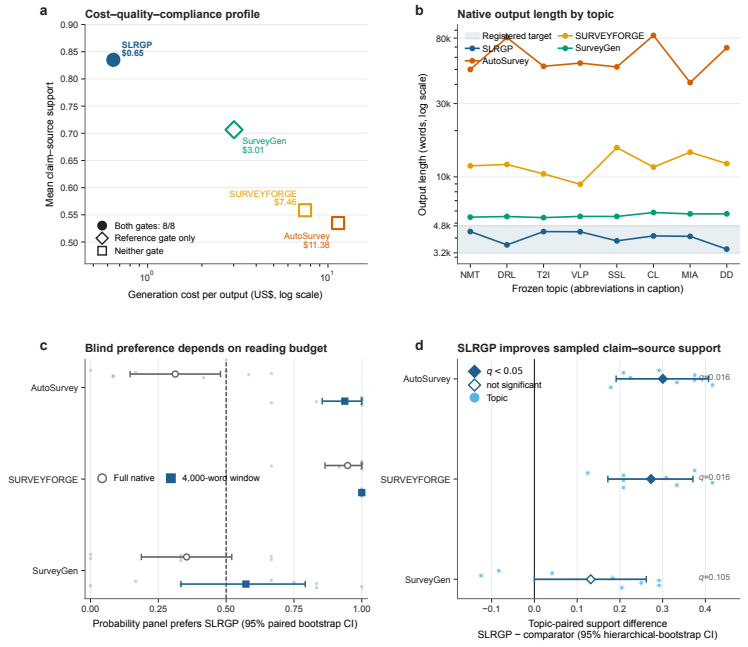


Figure 4: Native workflow length, generation-cost, sampled-support, and preference profiles. The panel comparison uses matched 4,000-word windows alongside each system's native cost and support measurements.

## G Evaluation Instrument and Model Roles

We tested the evaluation panel on two published reviews with eight controlled changes, three judges, and two repeated judgments, giving 96 valid calls. Shuffling sections reduced the organization score by 1.58 ($p = 0.0039$). Shuffling or replacing citations reduced citation plausibility by 2.67 and 1.42.

Adding redundant text reduced global coherence by 1.33. Deleting citations changed the score by $-0.08$ ($p = 1.0$), and removing synthesis language changed critical synthesis by $-0.33$ ($p = 0.375$). Krippendorff's $\alpha$ ranged from 0.26 to 0.47 across six raters. Based on these calibration results, the main analysis compares methods within the same topic and combines document-level panel scores with a separate claim–source support check. The strongest calibrated sensitivities concern organization, global coherence, and citation validity, which anchor the corresponding interpretations in the main text.

Model roles were separated. One fixed model labelled corpus structure and ran the representation test; a second generated all study outputs; an open-weight model provided the robustness check; and three other model families evaluated anonymized outputs. SHA-256 hashes determined splits and seeds, and every protocol change was logged with its reason and effect.