GANGMU: QUOTIENT-BASED CONSOLIDATION OF MATHEMATICAL KNOWLEDGE ACROSS DOCUMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Mathematical statements often appear in different forms across documents, yet systematically identifying and consolidating equivalent statements remains underexplored. We formulate Mathematical Knowledge Consolidation (MKC) using the Quotient-Based Graph Structure, which maps document-level statement nodes to canonical mathematical entities under strict semantic equivalence while preserving source evidence. Building on this structure, we introduce Gangmu, a two-stage framework that constructs an Evidence Graph of statements and relations and recovers a Canonical Graph through Mathematical Entity Resolution (MathER) in an open-world setting that permits unmatched statements. Gangmu uses semantic normalization, multi-evidence representations, uncertainty handling, and consistency checks to restrict linking and merging to confirmed equivalence. We also introduce MathER-Alg, a benchmark built from 50 algebra texts, comprising a reference knowledge base and 512 expert-annotated queries for open-world mathematical entity resolution. Experiments on MathER-Alg show that Gangmu achieves the highest overall accuracy among the evaluated methods.

1 INTRODUCTION

AI for mathematics has advanced rapidly, with substantial progress in mathematical problem solving and theorem proving (Trinh et al., 2024; Hubert et al., 2026). Several approaches in this area build on retrieving and reusing existing mathematical knowledge: retrieval-augmented provers and recent premise retrieval systems draw on existing definitions and theorems to support proof construction (Yang et al., 2023; Chen et al., 2025; Gao et al., 2026). Mathematical research agents also retrieve results from mathematical corpora and formal libraries (He et al., 2026; Ju et al., 2026) and accumulate intermediate findings in reusable knowledge stores (Liu et al., 2026; Kripner & Straka, 2026). Organizing mathematical statements together with their dependencies provides a structured basis for such retrieval and reuse, motivating research on graph-based representations of mathematical knowledge.

Mathematical knowledge graphs provide such representations, organizing statements and their dependencies into linked resources (Tomaszuk et al., 2023; Mishra et al., 2024). Recent efforts also address multiple sources: AutoMathKG uses retrieval and model judgments to fuse incoming entities, while TheoremGraph links formal and informal statements (Bian et al., 2025; Kurgan et al., 2026). Connected Theorems further highlights the need to consolidate different restatements of the same mathematical result into a single graph node (Bérczi et al., 2025). However, reliable consolidation requires distinguishing strictly equivalent statements from related but mathematically distinct ones. Systematic methods and evaluations for this decision across documents remain limited.

We study Mathematical Knowledge Consolidation (MKC): the problem of unifying mathematical statements, such as definitions and theorems, and their relations across documents while preserving their mathematical meaning. This task presents several challenges. Semantically equivalent statements may use different notation, terminology, or local definitions (Ye et al., 2026). Conversely, surface similarity does not imply semantic equivalence, since differences in domains, assumptions, quantifiers, or conclusions can change the mathematical content. Special cases and conditional variants must therefore remain distinct when they change the mathematical scope or content of the original statement (Zhang et al., 2026). Reliable consolidation must go beyond isolated statements to recover and preserve the relations among them (Mishra et al., 2024). Moreover, a mathematical

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.

statement under consideration may have no equivalent in the existing knowledge base, requiring an open-world setting that accommodates previously unrepresented entities. Reliable consolidation must therefore identify semantic equivalence across documents, preserve structural relations among statements, and allow statements without confirmed equivalents to remain separate.

To formulate this task, we introduce a quotient-based graph structure that connects statement nodes from individual documents to canonical mathematical entities. The Evidence Graph retains each extracted statement as a distinct source-linked node, together with its document-level relations. Strict semantic equivalence partitions evidence nodes into equivalence classes, each defining a canonical entity. Relations between evidence nodes are projected to their corresponding canonical entities to form the Canonical Graph. Retaining both graphs and their connecting mapping preserves the original statements and contexts, allowing multiple source formulations to support a single canonical entity.

We introduce Gangmu,[1] a two-stage framework for constructing this structure from mathematical documents (Figure 1). It first constructs the Evidence Graph through structure-aware document processing and extraction of mathematical statements and relations. It then performs Mathematical Entity Resolution (MathER) in an open-world setting to estimate the evidence-to-canonical mapping. Semantic normalization clarifies context-dependent expressions, while multiple evidence statements associated with each canonical candidate support reranking and semantic verification. Linking and merging require positive equivalence judgments and, where applicable, successful consistency checks; unresolved cases remain separate. Canonical content and relations are then assembled while preserving the underlying Evidence Graph.

We evaluate Gangmu through complementary component-level experiments. Evidence Graph correctness evaluation assesses the extracted statements and relations that underpin Canonical Graph recovery. Since recovery also depends on accurate Mathematical Entity Resolution (MathER), we introduce MathER-Alg to evaluate this core component in an open-world setting. Constructed from 50 algebra texts with a document-level split between reference-KB and query sources, the benchmark comprises a reference KB and 512 expert-annotated queries. On MathER-Alg, Gangmu achieves 86.33% overall accuracy, outperforming all six baselines. We also conduct four independent ablations to examine the contributions of MathER components.

We summarize our main contributions as follows:

1. We define Mathematical Knowledge Consolidation using a quotient-based graph structure that maps semantically equivalent statement nodes to shared canonical entities while retaining their source evidence and document-level relations.
2. We introduce Gangmu, a two-stage framework that constructs an Evidence Graph and recovers a Canonical Graph through open-world MathER, combining semantic normalization, multi-evidence candidate representations, and conservative resolution.
3. We present MathER-Alg, an expert-annotated benchmark constructed from 50 algebra texts, with a document-level split and 512 queries for evaluating equivalence linking and open-world NIL detection.

2 Related Work

Mathematical Knowledge Graphs. Mathematical knowledge graphs (KGs) organize mathematical objects, definitions, theorems, proofs, and their dependencies into linked, machine-readable structures. Structured mathematical knowledge formalisms and platforms such as OMDoc, MMT, and MathHub established foundations for interoperable representations (Kohlhase, 2006; Rabe & Kohlhase, 2013; Iancu et al., 2014). Direct mathematical KG efforts pursue different goals: MathGraph organizes concepts and relations for exercise solving, TheoremKB studies the extraction of results and proof dependencies, and MMLKG exposes definitions, statements, proofs, and dependencies from Mizar (Zhao et al., 2019; Mishra et al., 2024; Tomaszuk et al., 2023). More recently, AutoMathKG uses large language models (LLMs) and vector retrieval to extract and fuse mathematical entities, while TheoremGraph builds theorem-level dependency graphs and links formal

[1]The name Gangmu is inspired by Zheng Xuan's preface to the *Shi Pu*: "Lift the main cord of a net, and all its meshes open" (our translation).

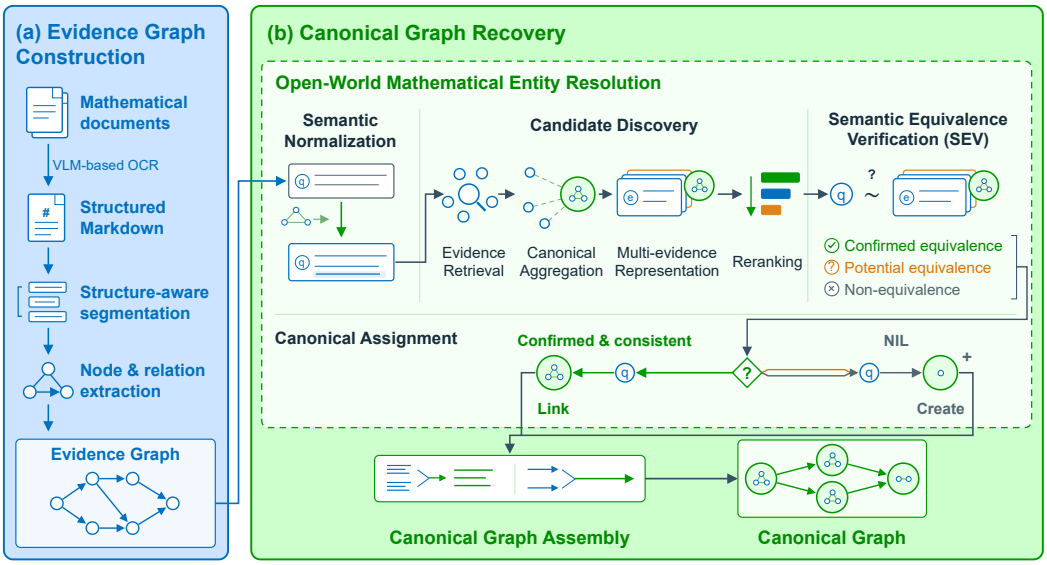

Figure 1: **Overview of Gangmu.** (a) Evidence graph construction extracts mathematical statements and document-level relations while preserving source provenance. (b) Canonical graph recovery performs open-world mathematical entity resolution through semantic normalization, candidate discovery, and semantic equivalence verification (SEV). Multi-evidence candidate representations support reranking and subsequent SEV. Confirmed, consistent matches permit linking; when no link is accepted (`NIL`), a separate canonical entity is created. Content synthesis and relation projection then assemble the Canonical Graph while retaining the underlying evidence. Blue circles denote evidence nodes, green circles canonical entities, and stacked cards multiple supporting evidence statements.

and informal statements (Bian et al., 2025; Kurgan et al., 2026). However, existing work primarily focuses on representation, extraction, or graph construction; dedicated methods and evaluations for consolidating semantically equivalent statements across documents remain limited.

Entity Resolution. Entity resolution (ER) identifies records or descriptions that refer to the same underlying entity and is a fundamental problem in data integration. Typical ER pipelines use blocking or candidate generation to limit comparisons, followed by matching and, when needed, clustering or collective inference (Christophides et al., 2021; Papadakis et al., 2020a). DeepER and DeepMatcher learn neural representations for matching (Ebraheem et al., 2018; Mudgal et al., 2018), Ditto fine-tunes pretrained language models (Li et al., 2020), and JedAI integrates ER stages into an end-to-end pipeline (Papadakis et al., 2020b). Related mathematical tasks include linking expressions to Wikipedia entities, recognizing formula concepts across equivalent representations, and linking related theorem statements (Kristianto et al., 2016; Scharpf et al., 2023; Kurgan et al., 2026). However, systematic entity resolution for mathematical knowledge units, such as definitions and theorems, remains comparatively underexplored.

3 Methodology

3.1 Problem Formulation

Mathematical Knowledge Consolidation (MKC) requires preserving document-level evidence while identifying mathematical statements that express the same underlying meaning across documents. We formulate this task using a quotient-based graph structure that connects statement nodes from individual documents to canonical mathematical entities.

Definition 1 (Quotient-Based Graph Structure)**.** For a given document collection $\mathcal{D}$, a Quotient-Based Graph Structure is a triple

$$\mathcal{Q}_{\mathcal{D}} = (G_E, G_C, \pi), \tag{1}$$

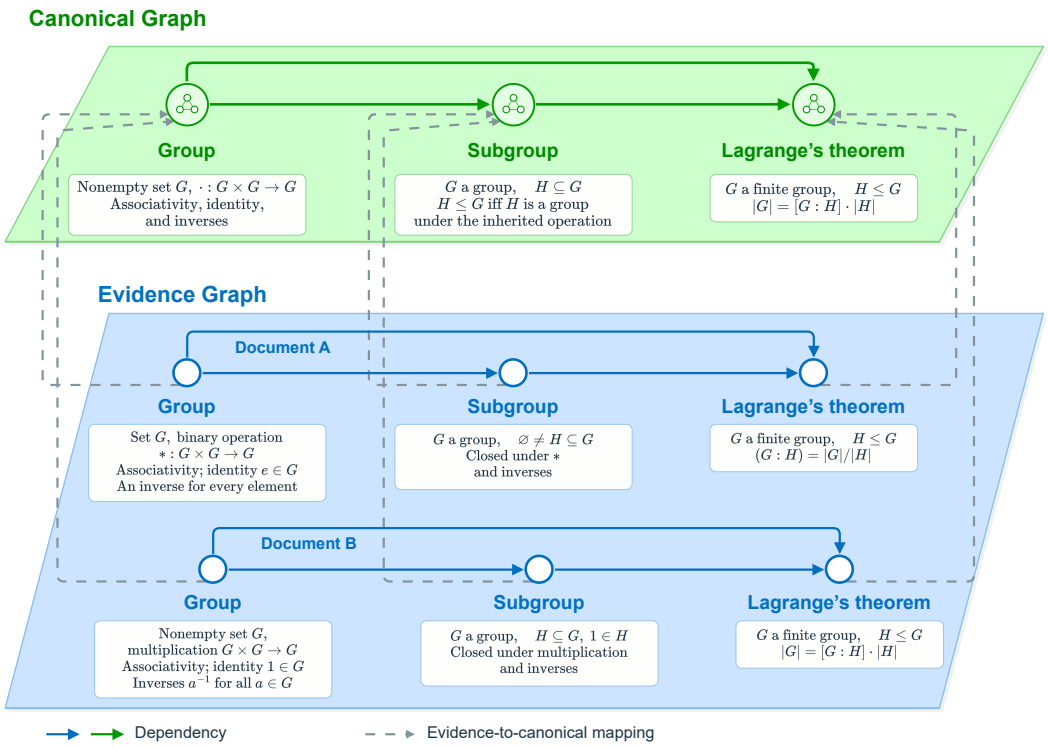

Figure 2: **An example of the quotient-based graph structure.** Blue nodes represent mathematical statements extracted from individual documents, and green nodes represent canonical entities. Gray dashed arrows show the evidence-to-canonical mapping; colored solid arrows denote dependencies within each layer. The connected circles inside canonical nodes symbolize equivalence classes, not their cardinalities. Mathematical statements in both layers are abbreviated for readability.

consisting of an Evidence Graph G_E, a Canonical Graph G_C, and an evidence-to-canonical mapping π, defined as follows:

1. **Evidence Graph.** For each document $d \in \mathcal{D}$, let V_d and E_d denote the node and edge sets of its document-level graph. The Evidence Graph is

$$V_E = \bigsqcup_{d \in \mathcal{D}} V_d, \qquad E_E = \bigsqcup_{d \in \mathcal{D}} E_d, \qquad G_E = (V_E, E_E). \tag{2}$$

   The disjoint union preserves document identity, so statement nodes from different documents remain distinct Evidence Nodes.

2. **Strict Semantic Equivalence and Quotient Projection.** Let $\sim$ denote the strict mathematical semantic equivalence relation on V_E:

$$u \sim v \iff \tau(u) = \tau(v) \ \wedge \ \mathrm{Sem}(u) = \mathrm{Sem}(v), \tag{3}$$

   where $\tau(\cdot)$ denotes the node type and $\mathrm{Sem}(\cdot)$ denotes a knowledge unit's full mathematical content, including domains, assumptions, quantifiers, conclusions, and definitional conventions, up to meaning-preserving changes in notation and wording. The resulting equivalence classes and induced projection are

$$V_C = V_E/\sim, \qquad \pi : V_E \to V_C, \qquad \pi(v) = [v]_\sim, \tag{4}$$

   where $[v]_\sim = \{u \in V_E : u \sim v\}$.

3. **Canonical Graph.** The Canonical Graph is

$$G_C = (V_C, E_C). \tag{5}$$

Each Evidence Edge (u, r, v) is projected to $(\pi(u), r, \pi(v))$. The resulting G_C is the quotient graph induced by $\sim$, with duplicate projected edges aggregated and self-loops omitted; all originating Evidence Edges remain in G_E.

The quotient graph G_C organizes canonical mathematical entities and their projected relations. The full structure $\mathcal{Q}_\mathcal{D}$ also retains G_E and the projection π from evidence nodes to canonical entities, preserving the extracted statements, document-level relations, and their canonical assignments. Figure 2 illustrates this structure using group and subgroup definitions and Lagrange's theorem: six statement nodes from two documents map to three canonical entities.

Definition 2 (Mathematical Knowledge Consolidation)**.** Given a collection $\mathcal{D} = \{d_1, \ldots, d_n\}$ of mathematical documents, where a document may be represented as PDF, Markdown, LaTeX, or any other format from which its mathematical content can be recovered, Mathematical Knowledge Consolidation (MKC) is the task of constructing a Quotient-Based Graph Structure:

$$\Phi(\mathcal{D}) = \mathcal{Q}_\mathcal{D} = (G_E, G_C, \pi). \tag{6}$$

MKC constructs the Evidence Graph G_E and recovers the Canonical Graph G_C together with the evidence-to-canonical mapping π. The task naturally decomposes into two stages:

$$\boxed{\mathcal{D} \xrightarrow{\text{Evidence Graph Construction}} G_E \xrightarrow{\text{Canonical Graph Recovery}} (G_C, \pi)}. \tag{7}$$

Evidence Graph Construction retains statement nodes and relations from individual documents without cross-document merging; Canonical Graph Recovery recovers (G_C, π) by resolving Evidence Nodes to canonical entities and projecting relations. Sections 3.2 and 3.3 instantiate these two stages.

3.2 Evidence Graph Construction

Gangmu first constructs the Evidence Graph G_E from the source corpus. We use GLM-5V-Turbo (GLM-5V-Turbo Team, 2026), a vision-language model, to parse PDF page images into structured Markdown, retaining mathematical expressions and the heading hierarchy. Structure-aware segmentation then follows this hierarchy to organize the document into manageable units. Sections exceeding the input budget are subdivided by paragraphs, with token-based splitting reserved for oversized paragraphs. Each segment retains its heading path and page locations, providing structural context for extraction and traceability to the source document.

An LLM extracts mathematical knowledge units from each segment as definition or theorem nodes. Theorem nodes encompass mathematical statements with labels such as lemma, proposition, and corollary, while preserving their original labels. Each node records its title, statement, source document, and location; theorem nodes additionally retain proofs when available. Extraction is instructed to preserve mathematical notation, assumptions, and conclusions. Proofs remain attached to their statements rather than forming a separate node type.

We recover document-internal relations through explicit references and candidate discovery over preceding nodes. Unambiguous references in statements or proofs establish corresponding edges; additional candidates are obtained through local proximity and lexical retrieval, then assessed by an LLM. A dependency relation records prior knowledge used in a statement, whereas a proof relation records knowledge used in its proof. Edges point from the prerequisite to the dependent node. Finally, the document graphs are assembled into G_E through the identity-preserving disjoint union defined in Section 3.1. Evidence Nodes from different documents remain distinct, retaining their statements, document-level relations, and provenance for subsequent canonical recovery. Further construction details are provided in Appendix E.

3.3 Canonical Graph Recovery

Given the Evidence Graph G_E, Gangmu recovers the Canonical Graph and the evidence-to-canonical mapping while retaining the extracted statements and document-level relations in G_E. Figure 1(b) outlines this process: Mathematical Entity Resolution (MathER) in an open-world setting establishes the evidence-to-canonical mapping, after which canonical content and relations are assembled.

3.3.1 QUOTIENT RECOVERY IN PRACTICE

The relation $\sim$ in Section 3.1 defines the ideal semantic partition of the evidence nodes. In practice, Gangmu relies on extracted statements and their context, a limited set of retrieved candidates, and model-based equivalence judgments, so exact recovery cannot generally be guaranteed. We therefore estimate the canonical partition from the available evidence.

Let $\widehat{V}_C$ denote the set of recovered canonical nodes. The final evidence-to-canonical mapping induces a partition:

$$\widehat{\pi} : V_E \longrightarrow \widehat{V}_C, \qquad u \mathrel{\widehat{\sim}} v \iff \widehat{\pi}(u) = \widehat{\pi}(v). \tag{8}$$

The relation $\widehat{\sim}$ groups Evidence Nodes assigned to the same canonical node and is therefore an equivalence relation. This structural property does not ensure strict semantic equivalence: the recovered groups may differ from the ideal classes defined by $\sim$. Over-merging groups inequivalent statements under one canonical node, potentially conflating incompatible statements and combining relations that should remain separate. Under-merging separates equivalent statements, retaining redundant canonical nodes without introducing a false equivalence. To preserve the semantic integrity of the canonical graph, Gangmu adopts a conservative resolution principle: linking and merging require confirmed equivalence and the applicable consistency checks, while unresolved cases retain separate canonical nodes.

3.3.2 OPEN-WORLD MATHEMATICAL ENTITY RESOLUTION

We consider an open-world setting in which an incoming Evidence Node may have no equivalent among the existing canonical nodes. Gangmu therefore supports both linking to an existing canonical node and creating a separate one. It processes documents in sequence, resolving each document's Evidence Nodes against the canonical nodes established by earlier documents. For a node q, it first prepares the statement for comparison across sources through semantic normalization. An LLM uses relevant dependency context when needed to clarify symbols, object domains, and local conventions while preserving the original assumptions, quantifiers, and conclusions. The resulting statement is retained alongside the source text and provides the query for embedding retrieval over previously assigned evidence of the same node type. Retrieving at the evidence level allows a canonical entity to be discovered through the different formulations used by its sources. Gangmu then groups the retrieved evidence nodes by their previously established canonical assignments to form a pool of canonical candidates. Hits with the same canonical assignment form one candidate, whose evidence supports subsequent equivalence judgments.

To assess whether q can be linked to any of these canonical candidates, Gangmu uses a multi-evidence candidate representation comprising normalized statements from center evidence, complementary representatives, and, when available, query-matched evidence. These statements provide alternative expressions of the candidate's mathematical content, reducing dependence on any single formulation. The representation supplies the candidate text for both reranking and verification. When the pool exceeds the candidate budget, a reranker compares q with these representations and retains the highest-ranked candidates; otherwise, all candidates proceed. For each retained candidate, an LLM then performs Semantic Equivalence Verification (SEV), comparing the normalized query with the candidate evidence under Equation 3. It examines domains, assumptions, quantifiers, conclusions, and definitional conventions; generalizations, special cases, and one-way implications do not establish strict semantic equivalence. To support conservative resolution, SEV distinguishes three outcomes: confirmed equivalence, potential equivalence, and non-equivalence. The intermediate outcome records unresolved semantic ambiguities without treating them as confirmed matches. Only confirmed equivalence can authorize linking or merging.

These judgments determine the canonical assignment of q. If no candidate is confirmed equivalent, Gangmu creates a separate canonical node; if exactly one is confirmed, it links q to that canonical node. When several candidates are confirmed, Gangmu checks their pairwise equivalence through SEV. Gangmu merges these canonical nodes and links q to the resulting canonical node only when every pair is confirmed equivalent and no unresolved merge conflict remains. Otherwise, q is assigned to a separate canonical node. Once the document's nodes have been assessed, Gangmu updates the canonical nodes and the recorded evidence-to-canonical assignments, making the resulting canonical state available to subsequent documents. Repeating this process yields the recovered mapping $\widehat{\pi}$ used for graph assembly.

Table 1: **Results of Gangmu and the baselines on MathER-Alg.** Boldface and underlining indicate the best and second-best scores for each metric, respectively. P denotes precision and R denotes recall.

Method	MathER-Alg				
	Linking P	NIL P	Linking R	NIL R	Overall Accuracy
(Baselines)					
Always-NIL	0.00%	54.10%	0.00%	**100.00%**	54.10%
Exact String Match	**100.00%**	54.31%	0.85%	**100.00%**	54.49%
Token Jaccard*	50.00%	61.48%	25.53%	87.00%	58.79%
BM25 + SEV	84.56%	71.35%	53.62%	<u>93.50%</u>	75.20%
Dense Retrieval*	65.32%	76.55%	61.70%	80.14%	71.68%
Claude Code Agent	72.66%	**91.48%**	**89.36%**	73.65%	<u>80.86%</u>
(Ours)					
Gangmu	<u>87.10%</u>	<u>85.76%</u>	<u>80.43%</u>	91.34%	**86.33%**

* Thresholds are selected to maximize overall accuracy on the MathER-Alg benchmark.

3.3.3 Canonical Graph Assembly

After canonical assignment, Gangmu synthesizes a canonical title and statement from member evidence. An LLM checks for changes to mathematical scope, assumptions, quantifiers, and conclusions and provides revision feedback. Individual normalization prepares statements for comparison, multi-evidence representations support equivalence judgments, and this synthesis consolidates the description of the established canonical entity. Each Evidence Edge (u, r, v) is projected to $(\widehat{\pi}(u), r, \widehat{\pi}(v))$. Duplicate relations are aggregated with their supporting evidence; projected self-loops are omitted while their original Evidence Edges remain. The recovered canonical nodes and projected relations form the Canonical Graph $\widehat{G}_C$. The extracted statements and document-level relations remain in G_E. The resulting quotient-based graph structure $\widehat{\mathcal{Q}}_{\mathcal{D}} = (G_E, \widehat{G}_C, \widehat{\pi})$ retains both evidence and the recovered canonical organization, keeping consolidated entities and relations traceable to their sources.

4 Experiments

4.1 The MathER-Alg Benchmark

MathER-Alg is a benchmark for Mathematical Entity Resolution (MathER) in algebra, comprising a reference knowledge base (KB) and an expert-annotated query set. Given a mathematical statement, a system must identify its canonical node in the reference KB under strict semantic equivalence, or return `NIL` if no such node exists. The source corpus comprises 50 expert-selected algebra texts, including textbooks and lecture notes, listed in Appendix G. We partition the corpus at the document level: 30 documents are used to construct the reference KB, while the remaining 20 serve as the source for queries, with no document contributing to both partitions.

The reference KB is initialized using the Gangmu pipeline with GPT-5.4 (OpenAI, 2026a). Experts then review potentially equivalent nodes and refine the graph. The resulting Canonical Graph contains 8,842 canonical nodes and 27,412 canonical edges. From the 20 query-source documents, experts curate and annotate 512 mathematical statements with tool assistance, retaining final authority over all labels. Of these queries, 235 have equivalent canonical nodes in the reference KB (in-KB queries), while 277 are labeled `NIL`. The reference KB and gold annotations are frozen for evaluation across all methods. Construction and annotation details are provided in Appendix B.

4.2 Open-World MathER Evaluation

4.2.1 Experimental Setting

Evaluation Protocol. All methods are evaluated on MathER-Alg; detailed protocols and configurations are provided in Appendix C. All methods requiring embeddings use Qwen3-Embedding-8B,

and all methods requiring reranking use Qwen3-Reranker-8B (Zhang et al., 2025). Gangmu, BM25 + SEV, and Claude Code Agent all use GPT-5.6-terra (OpenAI, 2026b).

Baselines. We compare Gangmu with six baselines:

- **Always-NIL.** This baseline returns `NIL` for every query.
- **Exact String Match.** Matches query and canonical statements after text normalization.
- **Token Jaccard.** Computes Jaccard similarity between the token sets of the query and canonical statements (Cohen et al., 2003). The threshold is selected to maximize overall accuracy on MathER-Alg.
- **BM25 + SEV.** Retrieves canonical statements with BM25 (Robertson & Zaragoza, 2009) and verifies candidates using Gangmu's SEV module.
- **Dense Retrieval.** Encodes the query and canonical statements and computes their cosine similarity. The threshold is selected to maximize overall accuracy on MathER-Alg.
- **Claude Code Agent.** Uses Claude Code (Anthropic, 2025) with Gangmu's evidence retrieval and reranking tools, and may request up to two additional searches with reformulated queries before selecting a canonical target or returning `NIL`.

Evaluation Metrics. Linking denotes assigning a query to a canonical target in the reference KB, whereas `NIL` denotes predicting that no strictly equivalent target exists. Our primary metric, **Overall Accuracy**, is the proportion of correctly resolved queries. Linking Precision and Recall measure links to gold canonical targets among all linking predictions and gold in-KB queries, respectively; NIL Precision and Recall measure correct NIL predictions among all NIL predictions and gold NIL queries, respectively.

4.2.2 Main Results

Table 1 reports the results of Gangmu and six baselines on MathER-Alg. We prioritize Overall Accuracy, which measures correct resolution across both in-KB and NIL queries. Gangmu achieves the highest accuracy of 86.33%, exceeding the strongest baseline, Claude Code Agent (80.86%), by 5.47 percentage points. BM25 + SEV, Dense Retrieval, and Token Jaccard reach 75.20%, 71.68%, and 58.79%, respectively. Gangmu maintains strong performance across all four precision and recall metrics (80.43%–91.34%), balancing successful linking with reliable NIL detection.

Compared with Claude Code Agent, Gangmu substantially improves Linking Precision (87.10% versus 72.66%) and NIL Recall (91.34% versus 73.65%). This pattern is consistent with Gangmu's conservative resolution principle: it favors well-supported links and more reliably rejects queries without an equivalent KB target. These gains come with lower NIL Precision (85.76% versus 91.48%) and Linking Recall (80.43% versus 89.36%), but the resulting balance delivers the highest overall accuracy on MathER-Alg.

4.3 Evidence Graph Correctness Evaluation

To assess the correctness of Gangmu's Evidence Graph, we sample across 30 source documents in proportion to the number of body pages in each document, obtaining fixed samples of 500 nodes and 500 edges. We use claude-opus-5 (Anthropic, 2026) as a third-party LLM judge, distinct from the model used in the Gangmu pipeline, to assign each item a binary correctness label. A node is correct only if its metadata is consistent with its extracted statement, the statement is mathematically valid under standard textbook conventions, and its normalized statement preserves the original mathematical meaning without substantive omissions. An edge is correct only if its endpoints, direction, relation type, and description are supported by the supplied node statements and, for proof relations, the target proof. We report the proportions of items judged correct as **Node Correctness** and **Edge Correctness**, obtaining **96.0%** (480/500) and **90.4%** (452/500), respectively.

To assess the reliability of this LLM judge, we sample 100 nodes and 100 edges from the respective 500-item samples for expert review. The judge's labels agree with the final expert labels in 96.0% of node cases and 95.0% of edge cases, showing high agreement with expert judgments on the reviewed

Table 2: **Ablation results of Gangmu on MathER-Alg.** Each ablation is applied independently to Gangmu. P and R denote precision and recall, respectively; w/o and w/ denote without and with. Boldface and underlining indicate the best and second-best scores for each metric, respectively.

Variant	Linking P	NIL P	Linking R	NIL R	Overall Accuracy
Gangmu	**87.10%**	85.76%	80.43%	**91.34%**	**86.33%**
w/o Semantic Normalization	74.13%	84.58%	<u>81.70%</u>	77.26%	79.30%
w/ Center-only Representation	78.95%	85.66%	**82.98%**	81.95%	82.42%
w/o Reranking	80.91%	**85.98%**	**82.98%**	84.12%	83.59%
w/ Binary SEV	<u>82.63%</u>	<u>85.87%</u>	**82.98%**	<u>85.56%</u>	<u>84.38%</u>

samples. Together, these results support the quality of the mathematical statements and relations in Gangmu's Evidence Graph.

4.4 Ablation Studies

We evaluate semantic normalization, multi-evidence candidate representation, candidate reranking, and explicit uncertainty handling in SEV through four independent ablations of Gangmu. All variants share the frozen reference KB, query set, and model settings, changing only the design under study. Table 2 reports the results; detailed settings are provided in Appendix F.

Without semantic normalization, raw statements replace normalized statements on both sides of retrieval, reranking, and SEV. Overall Accuracy falls from 86.33% to 79.30%, the largest drop among the ablations (7.03 percentage points). Linking Precision and NIL Recall also decline. These declines highlight semantic normalization as a major contributor to reliable cross-document linking, particularly in preventing incorrect links to the reference KB.

The center-only variant represents each canonical candidate using only its fixed center evidence for reranking and SEV, while preserving retrieval over the full Evidence Graph. Overall Accuracy falls to 82.42%, a decrease of 3.91 percentage points. Linking Precision and NIL Recall decline to 78.95% and 81.95%, respectively, suggesting that complementary evidence helps distinguish equivalent statements from misleadingly similar candidates and reduces incorrect links.

Without reranking, canonical candidates are ranked by embedding similarity under the same limit of 20 candidates. Overall Accuracy falls to 83.59%, a decrease of 2.73 percentage points. Linking Precision and NIL Recall also decline. These changes show the benefit of reranking retrieved candidates before semantic verification, reducing incorrect links without increasing the candidate limit.

Gangmu's SEV distinguishes confirmed equivalence, potential equivalence, and non-equivalence. The binary variant removes potential equivalence and assigns uncertain cases to non-equivalence, retaining the strict equivalence criterion and final linking rules. Overall Accuracy falls to 84.38%, a decrease of 1.95 percentage points. This pattern suggests that explicitly representing uncertainty during verification may help favor well-supported links. Although all four ablations increase Linking Recall, Gangmu achieves the highest Overall Accuracy, supporting the contributions of these components to balancing successful linking and reliable NIL detection.

5 Conclusion

Gangmu addresses Mathematical Knowledge Consolidation by organizing statement nodes and their dependencies in a Quotient-Based Graph Structure. Its two-stage pipeline first builds an Evidence Graph and then recovers a Canonical Graph through open-world MathER, preserving source evidence while restricting canonical assignments to confirmed equivalence. On MathER-Alg, Gangmu achieves 86.33% overall accuracy, the highest among the evaluated methods.

AI USE STATEMENT

We used generative AI tools to assist with polishing prose, finding related work, preparing preliminary drafts of selected sections, and writing and modifying parts of the codebase. The core methodology and experimental design were developed by the authors, who made all final scientific judgments. AI-generated drafts were treated as preliminary material and were reviewed, revised, and verified by the authors before inclusion in the paper. Generative AI also assisted with the construction of the MathER-Alg benchmark, including the conversion of mathematical documents from PDF to structured Markdown using a vision-language model. The benchmark's source material consists of existing mathematics textbooks and lecture notes. All queries and their annotations were reviewed and verified by domain experts, with revisions where necessary. The authors take full responsibility for the final content and claims.

ETHICS STATEMENT

In constructing MathER-Alg and planning its public release, we are committed to respecting the copyright and licensing terms of the source materials. The benchmark draws on 50 mathematics textbooks and lecture notes listed in Appendix G. Their terms include the GNU Free Documentation License, CC BY-SA, CC BY-NC, CC BY-NC-SA, CC BY-NC-ND, and author-specific conditions. Although existing licenses authorize reuse within their scope, separate permission has not been obtained for every source, and the scope of permitted redistribution remains uncertain for some materials.

The benchmark is intended for noncommercial research on mathematical entity resolution and will be made freely available to the research community. To limit redistribution of source expression, the release will omit original PDFs and proof text, which is not required for its statement-level resolution task. It will retain selected mathematical statements, annotations, graph structure, and source information for each item. After semantic normalization, many statements differ from their source formulations in wording or notation while retaining the same mathematical meaning. The resulting statements are organized across documents by semantic equivalence and dependency relations. This design focuses on mathematical facts and relationships while recognizing that their particular expression may remain protected by copyright. These measures are intended to reduce potential copyright risks and limit the benchmark's substitutability for the original textbooks.

Before public release, we will review source-derived content against applicable licenses and copyright rules, document the basis for redistribution, and retain required attribution and notices. Any license for our original annotations and documentation will exclude third-party material, which remains subject to its applicable terms. We will provide a contact channel for authors, other rights holders, and users to raise copyright or attribution concerns. We will promptly review such requests and, where appropriate, correct or remove affected material in accordance with applicable law.

REPRODUCIBILITY STATEMENT

We are committed to ensuring the reproducibility of our research. The Supplementary Material includes the complete frozen MathER-Alg benchmark, consisting of the two-layer KB and 512 queries with gold labels. Detailed results for the open-world MathER evaluation, ablation studies, and Evidence Graph correctness evaluation are also included in the Supplementary Material. An anonymous GitHub repository (`https://anonymous.4open.science/r/gangmu-2FEB/`) provides the source code for the complete Gangmu framework and all experiments, together with all prompt templates and supporting resources for reproducibility. Further details on Gangmu's methodology and graph construction are provided in Section 3 and Appendix E. Benchmark construction, annotation, and sources are described in Section 4.1 and Appendices B and G, while the evaluation protocol, baseline configurations, and ablation settings are detailed in Appendices C and F.

REFERENCES

Anthropic. Claude 3.7 Sonnet and Claude Code, February 2025. URL `https://www.anthropic.com/news/claude-3-7-sonnet`.

Anthropic. Introducing Claude Opus 5, July 2026. URL https://www.anthropic.com/news/claude-opus-5.

Gergely Bérczi, Bin Dong, Haocheng Ju, and Tianyi Xu. Connected Theorems: A graph-based approach to evaluating mathematical results, 2025. URL https://arxiv.org/abs/2508.17596.

Rong Bian, Yu Geng, Zijian Yang, and Bing Cheng. AutoMathKG: The automated mathematical knowledge graph based on LLM and vector database. *Computational Intelligence*, 41(4):e70096, 2025. doi: 10.1111/coin.70096.

Jiangjie Chen, Wenxiang Chen, Jiacheng Du, Jinyi Hu, Zhicheng Jiang, Allan Jie, Xiaoran Jin, Xing Jin, Chenggang Li, Wenlei Shi, Zhihong Wang, Mingxuan Wang, Chenrui Wei, Shufa Wei, Huajian Xin, Fan Yang, Weihao Gao, Zheng Yuan, Tianyang Zhan, Zeyu Zheng, Tianxi Zhou, and Thomas Hanwen Zhu. Seed-Prover 1.5: Mastering undergraduate-level theorem proving via learning from experience, 2025. URL https://arxiv.org/abs/2512.17260.

Vassilis Christophides, Vasilis Efthymiou, Themis Palpanas, George Papadakis, and Kostas Stefanidis. An overview of end-to-end entity resolution for big data. *ACM Computing Surveys*, 53(6): 1–42, 2021. doi: 10.1145/3418896.

William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web*, pp. 73–78, 2003. URL https://www.isi.edu/websites/ijcai03/papers/Cohen-p.pdf.

Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq Joty, Mourad Ouzzani, and Nan Tang. Distributed representations of tuples for entity resolution. *Proceedings of the VLDB Endowment*, 11(11):1454–1467, 2018. doi: 10.14778/3236187.3236198.

Guoxiong Gao, Zeming Sun, Jiedong Jiang, Yutong Wang, Jingda Xu, Peihao Wu, Bryan Dai, and Bin Dong. LeanSearch v2: Global premise retrieval for Lean 4 theorem proving, 2026. URL https://arxiv.org/abs/2605.13137.

GLM-5V-Turbo Team. GLM-5V-Turbo: Toward a native foundation model for multimodal agents, 2026. URL https://arxiv.org/abs/2604.26752.

Yutong He, Daibo Li, Guohong Li, Jiahe Geng, Zhengyang Huang, Can Ren, Zekun Zhang, Yifan Liu, Shuchen Zhu, Hengrui Zhang, Boao Kong, Ming Sun, Shu Li, Chenyi Li, Jiang Hu, Kun Yuan, Zaiwen Wen, and Pingwen Zhang. ReasFlow: Assisting reasoning-centric scientific discovery in applied mathematics via a knowledge-based multi-agent system, 2026. URL https://arxiv.org/abs/2607.14178.

Thomas Hubert, Rishi Mehta, Laurent Sartran, et al. Olympiad-level formal mathematical reasoning with reinforcement learning. *Nature*, 651:607–613, 2026. doi: 10.1038/s41586-025-09833-y. URL https://www.nature.com/articles/s41586-025-09833-y.

Mihnea Iancu, Constantin Jucovschi, Michael Kohlhase, and Tom Wiesing. System description: MathHub.info. In *Intelligent Computer Mathematics*, volume 8543 of *Lecture Notes in Computer Science*, pp. 431–434. Springer International Publishing, 2014. doi: 10.1007/978-3-319-08434-3_33.

Haocheng Ju, Guoxiong Gao, Jiedong Jiang, Bin Wu, Zeming Sun, Shurui Liu, Leheng Chen, Yutong Wang, Yuefeng Wang, Zichen Wang, Wanyi He, Peihao Wu, Liang Xiao, Ruochuan Liu, Bryan Dai, and Bin Dong. Automated conjecture resolution with formal verification, 2026. URL https://arxiv.org/abs/2604.03789.

Michael Kohlhase. *OMDoc – An Open Markup Format for Mathematical Documents [version 1.2]*, volume 4180 of *Lecture Notes in Computer Science*. Springer, 2006. doi: 10.1007/11826095.

Matěj Kripner and Milan Straka. OpenProver: Agentic and interactive theorem proving with Lean 4. In *19th Conference on Intelligent Computer Mathematics (CICM 2026)*, 2026. URL https://arxiv.org/abs/2607.09217.

Giovanni Yoko Kristianto, Goran Topić, and Akiko Aizawa. Entity linking for mathematical expressions in scientific documents. In *Digital Libraries: Knowledge, Information, and Data in an Open Access Society*, volume 10075 of *Lecture Notes in Computer Science*, pp. 144–149. Springer International Publishing, 2016. doi: 10.1007/978-3-319-49304-6_18.

Simon Kurgan, Evan Wang, Eric Leonen, Sophie Szeto, Luke Alexander, Artemii Remizov, Jarod Alper, Giovanni Inchiostro, and Vasily Ilin. TheoremGraph: Bridging formal and informal mathematics, 2026. URL https://arxiv.org/abs/2606.25363.

Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. Deep entity matching with pre-trained language models. *Proceedings of the VLDB Endowment*, 14(1):50–60, 2020. doi: 10.14778/3421424.3421431.

Jihao Liu, Guoxiong Gao, Zeming Sun, Bin Wu, Shurui Liu, Jiedong Jiang, Haocheng Ju, Leheng Chen, Ronnie Cheng, Xiping Zhang, and Bin Dong. Danus: Orchestrating mathematical reasoning agents with fact-graph memory, 2026. URL https://arxiv.org/abs/2607.06447.

Shrey Mishra, Yacine Brihmouche, Théo Delemazure, Antoine Gauquier, and Pierre Senellart. First steps in building a knowledge base of mathematical results. In *Proceedings of the Fourth Workshop on Scholarly Document Processing*, pp. 165–174. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.sdp-1.16.

Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data*, pp. 19–34, 2018. doi: 10.1145/3183713.3196926.

OpenAI. GPT-5.4 Thinking System Card, March 2026a. URL https://openai.com/index/gpt-5-4-thinking-system-card/.

OpenAI. GPT-5.6 Terra Model, 2026b. URL https://developers.openai.com/api/docs/models/gpt-5.6-terra. OpenAI API documentation. Accessed September 10, 2026.

George Papadakis, Dimitrios Skoutas, Emmanouil Thanos, and Themis Palpanas. Blocking and filtering techniques for entity resolution: A survey. *ACM Computing Surveys*, 53(2):1–42, 2020a. doi: 10.1145/3377455.

George Papadakis, Leonidas Tsekouras, Emmanouil Thanos, George Giannakopoulos, Themis Palpanas, and Manolis Koubarakis. Domain- and structure-agnostic end-to-end entity resolution with JedAI. *ACM SIGMOD Record*, 48(4):30–36, 2020b. doi: 10.1145/3385658.3385664.

Florian Rabe and Michael Kohlhase. A scalable module system. *Information and Computation*, 230:1–54, 2013. doi: 10.1016/j.ic.2013.06.001.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/1500000019. URL https://doi.org/10.1561/1500000019.

Philipp Scharpf, Moritz Schubotz, Howard S. Cohl, Corinna Breitinger, and Bela Gipp. Discovery and recognition of formula concepts using machine learning. *Scientometrics*, 128(9):4971–5025, 2023. doi: 10.1007/s11192-023-04667-9.

Dominik Tomaszuk, Łukasz Szeremeta, and Artur Korniłowicz. MMLKG: Knowledge graph for mathematical definitions, statements and proofs. *Scientific Data*, 10(1):791, 2023. doi: 10.1038/s41597-023-02681-3.

Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625:476–482, 2024. doi: 10.1038/s41586-023-06747-5. URL https://www.nature.com/articles/s41586-023-06747-5.

Kaiyu Yang, Aidan Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan J. Prenger, and Animashree Anandkumar. LeanDojo: Theorem proving with retrieval-augmented language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 21573–21612, 2023. doi: 10.52202/075280-0944. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/4441469427094f8873d0fecb0c4e1cee-Abstract-Datasets_and_Benchmarks.html.

Jiaying Ye, Samarth Rao, Leo Carlin, Kedar Chintalapati, Saharsh Bhargava, Rachit Jaiswal, Michael Zhou, Jared Darlington, Jiahe Lu, Jarod Alper, Vasily Ilin, and Henry Kvinge. Does my embedding reflect that $A = B$? Evaluating mathematical equivalence in embedding models, 2026. URL https://arxiv.org/abs/2606.23959.

Dechen Zhang, Xuan Tang, Xinxiang Yin, Xingwu Chen, Jian Qian, and Difan Zou. VALG: An agentic system for ML theory research, 2026. URL https://arxiv.org/abs/2608.13060.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 Embedding: Advancing text embedding and reranking through foundation models, 2025. URL https://arxiv.org/abs/2506.05176.

Tianyu Zhao, Yan Huang, Songfan Yang, Yuyu Luo, Jianhua Feng, Yong Wang, Haitao Yuan, Kang Pan, Kaiyu Li, Haoda Li, and Fu Zhu. MathGraph: A knowledge graph for automatically solving mathematical exercises. In *Database Systems for Advanced Applications*, volume 11446 of *Lecture Notes in Computer Science*, pp. 760–776. Springer International Publishing, 2019. doi: 10.1007/978-3-030-18576-3_45.

A QUOTIENT-BASED GRAPH SCHEMA

The following schemas specify how Gangmu represents the nodes and edges of the Quotient-Based Graph Structure introduced in Section 3. They describe the Evidence Graph and Canonical Graph records and the provenance links between these two layers. The tables list the core fields; supplementary metadata includes subject labels (`field`, `subfield`) and original statement labels where available. IDs are local to each node or edge collection. Dotted field names denote nested JSON fields, and brackets denote arrays.

Evidence Node

Field	Type	Description
`id`	integer	Evidence Node identifier.
`type`	string	`definition` or `theorem`.
`title`, `statement`	string	Extracted title and source statement.
`semantic.normalized_statement`	string	Statement normalized for comparison across documents.
`source_id`	string	Identifier of the source document.
`location`	object	`segment_id` and `pages`; `title_path` gives the heading path when available.
`in_edges`, `out_edges`	integer[]	Incoming and outgoing Evidence Edge IDs.

Evidence Edge

Field	Type	Description
`id`	integer	Evidence Edge identifier.
`source`, `target`	integer	Evidence Node IDs, directed from prerequisite to dependent.
`type`	string	`dependency` or `proof`.
`semantics`	string	Short description of the relation.

Canonical Node

Field	Type	Description
`id`	integer	Canonical Node identifier.
`type`	string	`definition` or `theorem`.
`title`, `statement`	string	Canonical description synthesized from member evidence.
`evidence_node_ids`	integer[]	Members of the recovered class $\widehat{\pi}^{-1}(\texttt{id})$.
`center_evidence_node_id`	integer	Fixed center evidence for the canonical node.
`diverse_evidence_node_ids`	integer[]	Up to two complementary representative evidence nodes.
`in_edges`, `out_edges`	integer[]	Incoming and outgoing Canonical Edge IDs.

Canonical Edge

Field	Type	Description
`id`	integer	Canonical Edge identifier.
`source`, `target`	integer	Canonical Node IDs after projection by $\widehat{\pi}$, preserving edge direction.
`types`	string[]	Relation types supported by the projected evidence.
`evidence_edge_ids`	integer[]	IDs of all Evidence Edges supporting this ordered endpoint pair.

Each Evidence Node belongs to exactly one Canonical Node, whose center and complementary representatives are selected from its members. Query-matched evidence is selected at runtime rather than stored as a fixed representative. Canonical Edge records aggregate an ordered pair of distinct endpoints. The released benchmark omits standalone `proof` text fields to limit redistribution of textbook proofs, but retains `proof`-type relations.

B MathER-Alg Construction and Annotation

This appendix describes reference KB curation and query annotation for MathER-Alg (Section 4.1). The source documents and their roles in the document-level split are listed in Appendix G.

Reference KB Construction and Review. The 30 reference-source documents were processed by Gangmu using GPT-5.4, yielding an initial Canonical Graph with 8,876 nodes. Experts reviewed 189 flagged pairs of potentially equivalent canonical nodes. The final decisions kept 149 pairs separate and merged 40 pairs, with one initial decision revised during conflict resolution. Applying these decisions consolidated 29 groups of nodes, reducing the node count by 34 to 8,842.

Model-Assisted Stratified Sampling. The Evidence Graph constructed from the 20 query-source documents contains a large pool of mathematical statements, making exhaustive manual annotation impractical. A uniform random sample would be expected to contain mostly NIL queries and relatively few linking queries. Obtaining sufficient linking examples in this way would therefore require substantial manual effort. We used model-assisted stratified sampling to obtain more potentially linkable statements while reducing annotation costs.

Specifically, we applied the MathER resolution procedure from Gangmu's Canonical Graph recovery stage with three models: GPT-5.4, GPT-5.6-sol, and DeepSeek-v4-pro. Each model produced provisional resolution outcomes against the reference KB. For sampling, we considered three groups: (i) statements for which all three models returned `NIL`; (ii) statements for which all three models predicted a link to the same canonical node; and (iii) statements for which the models disagreed on whether to link or which single canonical node to select.

We randomly sampled statements within each group, pooled the samples, and shuffled them before expert annotation. Annotators were not shown the models' provisional decisions, predicted canonical targets, or sampling-group assignments. They assigned labels anew through the human review and search procedure described below. The preliminary model outputs guided sampling but were not presented as label suggestions or adopted as gold annotations. This blinding was intended to prevent the preliminary model judgments from anchoring the experts' labeling decisions.

Human Annotation and NIL Verification. The annotation team comprised one mathematics instructor specializing in algebra and several mathematics graduate students. Each sampled query was independently assessed by two team members. Initial candidate retrieval returned up to 50 canonical nodes using Qwen3-Embedding-8B.

Annotators examined the candidates individually under the strict semantic equivalence criterion in Equation 3. A unique confirmed equivalent target supported a linking label. When the initial candidates did not resolve a query, annotators could search the full reference KB using freely formulated search text, without being restricted to the original query statement. A query was labeled `NIL` only

when no strictly equivalent target was identified after candidate inspection and additional searches. Queries for which annotation identified multiple equivalent canonical nodes, inadequate statement quality, or unresolved reviewer disagreements were excluded. The released reference KB and gold labels remain fixed across all evaluated methods. Although missed equivalents and residual KB errors remain possible, we sought to make the benchmark as reliable as possible through independent expert annotation, additional searches, and careful exclusion of unresolved cases.

C Open-World MathER Evaluation Details

Evaluation Protocol. All methods are evaluated independently on the same frozen reference KB and 512-query set introduced in Section 4.1. The associated Evidence Graph, evidence-to-canonical mapping, and gold annotations remain fixed; resolving a query does not update the KB. Content-based methods receive normalized query statements, and retrieval is restricted to nodes of the same type as the query. Evaluation uses the metrics in Section 4.2; precision is reported as zero when no predictions of the corresponding type are made.

Gangmu Configuration. Gangmu retrieves up to 50 evidence nodes of the same type with the shared embedder and aggregates these hits into canonical candidates through the evidence-to-canonical mapping. Candidates use the multi-evidence representation described in Section 3.3. If more than 20 candidates are available, the shared reranker retains the highest-ranked 20; otherwise, all candidates proceed to SEV. The default Gangmu configuration is summarized in Table 3.

Table 3: Default configuration of Gangmu for open-world MathER evaluation.

Setting	Configuration
Verification model	GPT-5.6-terra
Embedding model	Qwen3-Embedding-8B
Embedding representation	2048-dimensional, ℓ_2-normalized
Reranking model	Qwen3-Reranker-8B
Initial retrieval	Up to 50 evidence nodes of the same type
Verification candidate limit	20 canonical candidates

Linking Rules and Sensitivity Analysis. When multiple candidate canonical entities are judged equivalent to a query, Gangmu additionally checks every distinct pair of these candidates using the pairwise SEV rule in Section 3.3.2. Their existing canonical IDs are returned as an accepted set only if every candidate is confirmed equivalent to every other candidate; otherwise, Gangmu returns `NIL`, without modifying the frozen KB. Each accepted set counts as one linking prediction, which is correct if and only if it contains the gold canonical ID. In the main Gangmu run, only 2 of the 512 queries (0.39%) return accepted sets containing multiple canonical IDs; both predictions are correct under the original rule. We re-score the saved predictions under a single-target constraint by replacing these two outputs with `NIL` and leaving all others unchanged. As shown in Table 4, Overall Accuracy decreases from 86.33% to 85.94%, remaining 5.08 percentage points above Claude Code Agent, the strongest baseline (80.86%). All six baselines already return at most one canonical ID per query, so their scores are unchanged under this constraint. These results show that Gangmu's advantage over the evaluated baselines is robust to conservatively treating all multi-ID outputs as `NIL`.

Table 4: Overall Accuracy under the original and single-target decision rules for open-world MathER evaluation. The single-target constraint replaces accepted sets containing multiple canonical IDs with `NIL`. The two affected queries are 282 and 365.

Method	Correct queries	Overall Accuracy
Claude Code Agent	414/512	80.86%
Gangmu (original)	442/512	86.33%
Gangmu (single-target)	440/512	85.94%

Baseline Configurations.

- **Always-NIL.** This baseline returns `NIL` for every query.
- **Exact String Match.** We match the query against canonical statements after Unicode NFKC normalization, lowercasing, and whitespace removal, returning the matching node or `NIL`. Ties are resolved by the smallest canonical node ID.
- **Token Jaccard.** We compute Jaccard similarity (Cohen et al., 2003) between the token sets of the query and canonical statements, linking to the highest-scoring node if its score meets the threshold and returning `NIL` otherwise. The threshold is selected to maximize overall accuracy on the MathER-Alg benchmark.
- **BM25 + SEV.** BM25 (Robertson & Zaragoza, 2009) retrieves 20 canonical statements with $k_1 = 1.5$ and $b = 0.75$. We reuse Gangmu's Semantic Equivalence Verification (SEV) module without modification to verify candidates in rank order using their associated evidence, returning the first equivalent candidate or `NIL` if none qualifies.
- **Dense Retrieval.** Using the shared embedder, we rank canonical statements by cosine similarity to the query, linking to the highest-scoring node if its score meets the threshold and returning `NIL` otherwise. The threshold is selected to maximize overall accuracy on the MathER-Alg benchmark.
- **Claude Code Agent.** Claude Code (Anthropic, 2025) (v2.1.220) runs an agent using Gangmu's evidence retrieval and reranking together with the associated candidate evidence. It receives up to 20 candidates per search and may request two additional searches with reformulated queries before returning one canonical node or `NIL`. As specified in Section 4.2, the agent uses GPT-5.6-terra, the same LLM as Gangmu and BM25 + SEV.

D GANGMU CASE STUDIES

We present two MathER-Alg examples illustrating Gangmu's linking and `NIL` decisions. Each box shows the original query, its gold label, and Gangmu's prediction, together with the matched canonical statement or a retrieved candidate rejected by SEV. The analysis following each box relates these statements to the recorded retrieval and verification steps. As in the benchmark files, `MERGE` denotes a link to an existing canonical node; evaluation does not modify the frozen KB.

MathER-Alg Query 321: Equivalence under Variable Renaming

Query. Let G be a group. We let

$$Z(G) = \{z \in G : az = za \text{ for all } a \in G\}.$$

$Z(G)$ is called the center of G.

Gold label. `MERGE`, Canonical Node 200.

Gangmu prediction. `MERGE`, Canonical Node 200.

Matched canonical statement (Node 200). For a group G, the center of G is the set

$$Z(G) = \{x \in G : xg = gx \text{ for all } g \in G\}.$$

Analysis. Both statements define the center of an arbitrary group. Renaming z to x and a to g turns the query's condition $az = za$ into $gx = xg$; exchanging the two sides of this equality gives $xg = gx$, exactly as in the canonical statement. The two definitions therefore specify the same subset of G, with identical domains and universal quantifiers.

In the recorded run, the six highest-ranked evidence hits all belonged to Canonical Node 200 and were aggregated into one canonical candidate. This candidate ranked second after reranking, among the 20 retained for query-to-candidate SEV. SEV judged it equivalent and the other 19 candidates non-equivalent, so Gangmu returned the unique confirmed target. This example illustrates how the quotient-based structure groups repeated definitions across documents and how semantic verification confirms their strict semantic equivalence despite minor notational differences.

MathER-Alg Query 135: Similar Formulas with Different Scalar Domains

Query. Suppose that $B = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \ldots, \mathbf{v}_p\}$ is an orthonormal basis of the subspace W of $\mathbb{C}^m$. For any $\mathbf{w} \in W$,

$$\mathbf{w} = \langle \mathbf{v}_1, \mathbf{w} \rangle \mathbf{v}_1 + \langle \mathbf{v}_2, \mathbf{w} \rangle \mathbf{v}_2 + \langle \mathbf{v}_3, \mathbf{w} \rangle \mathbf{v}_3 + \cdots + \langle \mathbf{v}_p, \mathbf{w} \rangle \mathbf{v}_p.$$

Gold label. `NIL`.

Gangmu prediction. `NIL`.

Retrieved candidate rejected by SEV (Canonical Node 1414). Let W be a real inner product space with orthonormal basis $\{w_1, \ldots, w_n\}$. Then for every $v \in W$,

$$v = w_1(v \cdot w_1) + \cdots + w_n(v \cdot w_n).$$

Analysis. Both statements express an orthonormal-basis expansion, but their scalar domains differ: the query concerns a complex vector space, whereas the displayed candidate is restricted to a real inner product space. The candidate's center evidence, Evidence Node 1546, originally abbreviates its setting as "IPS." Its dependency definitions explicitly specify a real vector space; semantic normalization carries this contextual restriction into the statement used for comparison. The domain distinction is therefore visible to SEV rather than left implicit in the source's terminology.

This candidate was not missed by retrieval: its evidence ranked first, and the canonical candidate ranked third after reranking. SEV nevertheless judged it non-equivalent. All other 19 retained candidates also received non-equivalence judgments, leading Gangmu to return `NIL`, in agreement with the gold label. This example illustrates why formula similarity alone is insufficient for strict equivalence when the scalar domains differ.

E Evidence Graph Construction Details

E.1 Document Parsing and Segmentation

Each PDF page is rendered as an image and processed by GLM-5V-Turbo (GLM-5V-Turbo Team, 2026). The parsing instructions request structured Markdown for the current page, preserving mathematical expressions, headings, statement labels, and a source-page marker. Page outputs are checked for empty content, malformed mathematical delimiters, inconsistent LaTeX environments, and abnormal repetition, then assembled in page order. Document-level cleanup removes repeated headers and footers and repairs obvious inconsistencies in the heading hierarchy while preserving the mathematical content.

Segmentation recursively follows Markdown heading levels from coarse to fine. Sections exceeding the input budget are split at deeper headings where available; paragraph packing is used when the heading hierarchy provides no further subdivision. Only paragraphs that individually exceed the budget are split into token windows. Each segment retains its position in the document, the path of enclosing headings, and its source pages. These records provide the extraction model with section context and support locating the resulting nodes in the source document.

E.2 Node and Relation Extraction

We use an LLM for node extraction and model-based relation assessment. Extraction targets explicit definitions, labeled mathematical statements, and reusable mathematical objects or constructions. Exercises, examples, remarks, and intermediate proof claims are generally excluded as standalone nodes, as are unlabeled theorem-like statements. Nodes use the definition or theorem schema, with original labels such as lemma, proposition, and corollary retained. Each node records its title, statement, subject area, source document, and location; available proofs are attached to theorem nodes. The model is instructed to produce English prose while preserving notation, assumptions, and conclusions. Within each document, exact duplicates with the same type and statement after basic text cleanup are removed, and statements are checked for malformed mathematical syntax and apparent truncation.

Relation recovery considers only nodes preceding the target in the same document. Unambiguous label references in the target statement or proof directly establish dependency or proof relations, respectively. Additional candidates combine nearby preceding nodes with BM25 retrieval (Robertson & Zaragoza, 2009) using the target statement and, when available, its proof. Candidates already connected by explicit references are excluded from further assessment. For the remaining candidates, the model determines whether the target statement uses the candidate's mathematical content, the proof uses it, or neither relation is supported. These outcomes yield a dependency edge, a proof edge, or no edge. Dependency takes precedence when both relations apply to the same pair. Every edge points from the prerequisite to the dependent node and includes a short description of the relation.

Before assembly, we check edge endpoints, relation types, and consistency between edges and node adjacency records. Document graphs are then combined through the identity-preserving disjoint union in Section 3.1. This preserves source-specific nodes, attached proofs, document-internal edges, and provenance in the Evidence Graph used for subsequent canonical recovery.

F Ablation Study Details

F.1 Shared Experimental Settings

Each ablation compares a single modification with Gangmu on the same frozen MathER-Alg reference KB and 512-query set (Section 4.1). The associated Evidence Graph, evidence-to-canonical mapping, and gold annotations remain fixed. Evaluation follows the protocol and metrics in Section 4.2. Table 3 summarizes the default Gangmu configuration; the changes introduced by each ablation are described below. Gangmu applies reranking when more than 20 canonical candidates are available and otherwise retains all candidates.

F.2 Without Semantic Normalization

This variant replaces the normalized statement of each evidence node with its raw statement on both the query and candidate sides. The replacement covers embedding retrieval, reranking, query-to-candidate SEV, and any subsequent strict pairwise verification between candidate clusters. Embedding inputs and downstream mathematical comparisons therefore use the same raw text representation. The reference KB, Evidence Graph, and evidence-to-canonical mapping remain fixed. All other aspects of the pipeline remain unchanged from Gangmu.

F.3 Center-only Candidate Representation

Gangmu's multi-evidence candidate representation uses a fixed center evidence node together with up to two complementary representative nodes and, when available, query-matched evidence. Repeated evidence IDs and identical normalized statements are deduplicated. These evidence statements supply the candidate text for reranking and query-to-candidate SEV. Strict pairwise verification between candidate clusters uses their center and complementary representative evidence, without query-matched evidence.

The center-only variant retains only the fixed center evidence of each canonical candidate in all three stages. Each center is already recorded in the frozen reference KB and is independent of the current query. Both candidates in a strict pairwise comparison therefore use only their respective center evidence. Normalized statements and canonical metadata follow Gangmu's existing representation.

Evidence retrieval remains unchanged: both variants search the full Evidence Graph and aggregate hits into the same canonical candidate pool before reranking. Query text, model settings, prompts, candidate budgets, and final linking rules remain unchanged. The representation change can affect reranking scores and the retained candidates, as well as subsequent SEV decisions.

F.4 Without Reranking

This variant preserves evidence retrieval and aggregation into canonical candidates but omits the reranking step. A canonical candidate's score is the maximum cosine similarity between the query

and any of its retrieved evidence nodes. Candidates are ordered by decreasing score, with ties broken by ascending canonical node ID. At most 20 candidates are retained; all are kept when fewer than 20 are available. All other aspects of the pipeline remain unchanged from Gangmu.

F.5 BINARY SEV

Gangmu's SEV distinguishes confirmed equivalence, potential equivalence, and non-equivalence. The binary variant removes potential equivalence from the verification prompts and permitted outputs. A comparison is judged equivalent only when the available evidence establishes complete, strict semantic equivalence under Gangmu's criterion. Cases where equivalence cannot be confirmed, including uncertain cases, receive a non-equivalence judgment.

This change applies both to query-to-candidate verification and to the strict pairwise verification between candidate clusters when multiple candidates are judged equivalent. The relation definitions and output constraints are adjusted accordingly. All other aspects of the pipeline remain unchanged from Gangmu.

G MATHER-ALG DATA SOURCES

The source corpus spans a broad range of algebraic topics, including linear and abstract algebra, group and Galois theory, commutative algebra, number theory, representation theory, Lie theory, category theory, and algebraic geometry. The 50 source documents used in MathER-Alg are listed below, grouped by their roles in the document-level split.

G.1 REFERENCE KNOWLEDGE BASE SOURCES

The following 30 documents supply the reference KB.

1. Thomas W. Judson. *Abstract Algebra: Theory and Applications* (2025).
2. Frederick M. Goodman. *Algebra: Abstract and Concrete* (2015).
3. Edwin H. Connell. *Elements of Abstract and Linear Algebra* (2004).
4. Tom Denton. *Introduction to Algebraic Structures*.
5. Jim Hefferon. *Linear Algebra* (4th ed., 2020).
6. Stephen Boyd and Lieven Vandenberghe. *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares* (2018).
7. Sheldon Axler. *Linear Algebra Done Right* (4th ed., 2024).
8. Alexander Hulpke, with contributions by Kenneth Monks and Ellen Ziliak. *Abstract Algebra in GAP* (2011).
9. J. S. Milne. *Group Theory* (2025).
10. J. S. Milne. *Fields and Galois Theory* (2022).
11. William Stein. *Elementary Number Theory: Primes, Congruences, and Secrets* (2017).
12. Anthony W. Knapp. *Basic Algebra* (digital 2nd ed., 2016).
13. Stanley N. Burris and H. P. Sankappanavar. *A Course in Universal Algebra* (2012).
14. J. S. Milne. *A Primer of Commutative Algebra* (2020).
15. Robert B. Ash. *A Course in Commutative Algebra* (2003).
16. Kevin McGerty. *Algebra II: Rings and Modules* (2016).
17. Emily Riehl. *Category Theory in Context* (2nd ed.).
18. Pavel Etingof et al. *Introduction to Representation Theory* (2011).
19. Alexander Kirillov, Jr. *Introduction to Lie Groups and Lie Algebras*.
20. Victor Shoup. *A Computational Introduction to Number Theory and Algebra* (2007).
21. J. S. Milne. *Algebraic Number Theory* (2020).

22. J. S. Milne. *Algebraic Geometry* (2024).
23. David A. Craven. *Finite Group Theory* (2010).
24. J. S. Milne. *Algebraic Groups and Arithmetic Groups* (2006).
25. Irena Swanson. *Homological Algebra* (2024).
26. Roman Bezrukavnikov. *Noncommutative Algebra* (2023).
27. Alexander M. Kasprzyk. *Computational Commutative Algebra Notes*.
28. Ravi Vakil. *Foundations of Algebraic Geometry* (2012 draft).
29. Victoria Gould. *Semigroup Theory: A Lecture Course*.
30. Darij Grinberg and Victor Reiner. *Hopf Algebras in Combinatorics* (2017).

G.2 Query Set Sources

The following 20 documents supply the query set.

31. Dana C. Ernst. *An Inquiry-Based Approach to Abstract Algebra* (2016).
32. Jessica K. Sklar. *First-Semester Abstract Algebra: A Structural Approach*.
33. Sergei Treil. *Linear Algebra Done Wrong* (2014).
34. Robert A. Beezer. *A First Course in Linear Algebra* (2015).
35. Tom Leinster. *Galois Theory* (2024).
36. Michael Janssen and Melissa Lindsey. *Rings with Inquiry*.
37. Sergey Mozgovoy. *Fields, Rings and Modules* (2026).
38. George M. Bergman. *An Invitation to General Algebra and Universal Constructions*.
39. Pete L. Clark. *Commutative Algebra* (2013).
40. Tom Leinster. *Basic Category Theory* (2014).
41. Anthony W. Knapp. *Advanced Algebra* (digital 2nd ed., 2016).
42. Benjamin Steinberg. *Representation Theory of Finite Groups* (2009).
43. Anthony W. Knapp. *Lie Groups Beyond an Introduction* (digital 2nd ed., 2023).
44. Robert B. Ash. *A Course in Algebraic Number Theory* (2003).
45. Steffen Oppermann. *Homological Algebra* (2016).
46. Michael Artin. *Noncommutative Rings* (1999).
47. Clara Löh. *Geometric Group Theory* (2022).
48. Andreas Gathmann. *Algebraic Geometry* (2021–2022).
49. Yairon Cid-Ruiz. *Lecture Notes in Computer Algebra: A First Introduction to Computational Algebraic Geometry and Commutative Algebra*.
50. Ignacio López Franco (lectures) and David Mehrle (notes). *Topics in Category Theory: Hopf Algebras* (2016).

H Prompt Templates

This appendix presents four selected prompt templates central to Gangmu and its evaluation. The complete set of prompt templates is available in the anonymous repository described in the Reproducibility statement. Only presentation formatting is adjusted below.

H.1 SEMANTIC NORMALIZATION

This template performs semantic normalization for open-world MathER (Section 3.3.2). It uses an Evidence Node and relevant dependency context to produce a normalized statement for evidence retrieval and semantic comparison while preserving the original mathematical meaning.

Prompt Template for Semantic Normalization

Role
You are a professional mathematical semantic editor and logician.
Your task is to convert one mathematical knowledge-graph node into a compact, faithful, self-contained normalized statement for embedding retrieval and cross-document alignment.

Output
Return exactly one valid JSON object:

```
{
  "normalized_statement": ""
}
```

The object must contain exactly this one key. Do not return Markdown, comments, explanations, code fences, or additional fields.
Inside JSON strings, every LaTeX backslash must be escaped as \\. Never emit an invalid JSON escape such as \alpha; emit \\alpha.

Input
The input is one JSON object:

```
{
  "node": {
    "id": 12,
    "type": "definition or theorem",
    "title": "...",
    "statement": "..."
  },
  "dependencies": [
    {
      "id": 2,
      "type": "definition",
      "title": "...",
      "statement": "..."
    }
  ]
}
```

The current node's `statement` is authoritative. Dependencies contain only predecessors connected to the current node by dependency edges. They are optional context and may be irrelevant, incomplete, noisy, or wrong.

Core rules

1. Preserve the exact mathematical meaning of the current statement.
2. Correctness is more important than adding context.
3. Keep all essential assumptions, domains, parameters, quantifiers, and conclusions.
4. Do not strengthen or weaken the statement.
5. Do not invent facts from general mathematical knowledge.
6. Do not import proofs, examples, historical remarks, or consequences.
7. Preserve existing mathematical notation and LaTeX.
8. Remove filler phrases such as `Clearly`, `Obviously`, `Note that`, and `It is easy to see` when they add no mathematical content.

Prompt Template for Semantic Normalization (continued)

9. Keep the result compact but sufficiently self-contained for retrieval.
10. If the current statement is already concise and unambiguous, make only conservative formatting improvements.

Dependency rules
Use a dependency only when it directly helps to:

1. clarify an ambiguous symbol, object, or notation in the current statement;
2. make an implicit ambient object explicit;
3. restore a missing domain or parameter required to understand the statement;
4. expand a shorthand or document-specific convention.

Do not use a dependency:

1. merely because it is connected by an edge;
2. when it shares only broad terms such as group, ring, field, set, or map;
3. when it would add a new hypothesis or conclusion;
4. when it defines a different object;
5. when its relevance is uncertain.

Definitions
For a definition:

1. preserve the object, property, operation, relation, or notation being defined;
2. do not force the definition into theorem form;
3. make the ambient domain explicit only when supported;
4. retain notation introduced by the definition;
5. keep multiple inseparable parts of one definition together;
6. omit examples and explanatory paraphrases that are not part of the definition.

Theorems
For a theorem:

1. prefer a clear hypothesis-to-conclusion form;
2. use `Let ..., Then ...` when there are global hypotheses;
3. preserve local hypotheses inside their numbered parts;
4. preserve all essential conclusions;
5. do not promote variables occurring only inside quantified conclusions to global assumptions.

Examples
Input:

```
{
  "node": {
    "id": 12,
    "type": "theorem",
    "title": "Inverse of a product",
    "statement": "If $a$ and $b$ are in $G$, then
    $(ab)^{-1}=b^{-1}a^{-1}$."
  },
  "dependencies": [
    {
      "id": 2,
```

Prompt Template for Semantic Normalization (continued)

```
      "type": "definition",
      "title": "Group",
      "statement": "A group is a set $G$ with an associative
      binary operation, identity, and inverses."
    }
  ]
}
```

Output:

```
{
  "normalized_statement": "Let $G$ be a group and let $a,b \\in
  G$. Then $(ab)^{-1}=b^{-1}a^{-1}$."
}
```

Input:

```
{
  "node": {
    "id": 25,
    "type": "definition",
    "title": "Centralizer",
    "statement": "The centralizer of $a$ in $G$ is
    $C(a)=\\{g\\in G:ga=ag\\}$."
  },
  "dependencies": [
    {
      "id": 2,
      "type": "definition",
      "title": "Group",
      "statement": "A group is a set with an associative
      operation, identity, and inverses."
    }
  ]
}
```

Output:

```
{
  "normalized_statement": "In a group $G$, the centralizer of
  $a \\in G$ is $C(a)=\\{g\\in G:ga=ag\\}$."
}
```

H.2 Query-to-Candidate SEV

Gangmu uses this template for query-to-candidate Semantic Equivalence Verification (SEV), comparing the normalized query with the representative evidence of a canonical candidate (Section 3.3.2). BM25 + SEV reuses the same template (Appendix C).

Prompt Template for Query-to-Candidate SEV

Role
You are an expert research mathematician responsible for strict semantic alignment in a cross-document mathematical knowledge graph. Your decisions must be conservative: merge nodes only when their complete mathematical content is strictly equivalent.

Input
You receive exactly one JSON object with the following structure:

```
{
  "current_evidence_node": {
```

Prompt Template for Query-to-Candidate SEV (continued)

```
    "evidence_node_id": "integer",
    "type": "definition | theorem",
    "title": "string",
    "field": "string",
    "subfield": "string",
    "normalized_statement": "string"
  },
  "candidate_canonical_node": {
    "canonical_node_id": "integer",
    "type": "definition | theorem",
    "title": "string",
    "field": "string",
    "subfield": "string",
    "representative_evidence_nodes": [
      {
        "role": "center_evidence | diverse_evidence_1 |
        diverse_evidence_2 | query_matched_evidence",
        "evidence_node_id": "integer",
        "normalized_statement": "string"
      }
    ]
  }
}
```

`representative_evidence_nodes` contains one to four distinct representative Evidence Nodes:

- exactly one center Evidence Node;
- at most two diverse Evidence Nodes;
- optionally one Evidence Node selected as most similar to the current node.

Some optional roles may be absent or may refer to content already represented by another role and therefore be omitted.
Use only `normalized_statement` as the mathematical content for semantic comparison. `type`, `title`, `field`, and `subfield` provide context but do not establish equivalence. The current node and candidate have already passed the same-type filter.

Task
Determine whether the current Evidence Node belongs to the strict-equivalence class represented by the candidate Canonical Node.

1. Compare `current_evidence_node.normalized_statement` independently with the `normalized_statement` of every item in `candidate_canonical_node.representative_evidence_nodes`.
2. For each comparison, check the mathematical object or proposition, assumptions, ambient setting, scope, quantifiers, conclusions, and logical direction.
3. Aggregate the individual comparisons using the relation rules below.

Do not combine different representative statements into a new statement. Do not use majority voting. The current Evidence Node is strictly equivalent to the Canonical Node only when it is strictly equivalent to every supplied representative statement.

Relation rules

`equivalent`
Return `equivalent` if and only if the current `normalized_statement` is strictly equivalent to every representative `normalized_statement`:

- the same mathematical object or construction is defined, or the same proposition is asserted;

Prompt Template for Query-to-Candidate SEV (continued)

- assumptions, ambient setting, scope, quantifiers, conclusions, and logical direction agree;
- differences are limited to notation, variable names, order of presentation, standard definitional expansion, or logically equivalent reformulation.

Textual identity is not required. Mathematical equivalence under exactly the same assumptions is required.

`possible_equivalent`
Return `possible_equivalent` only when all of the following hold:

- no representative has an identifiable mathematical difference from the current node;
- at least one comparison is extremely close to strict equivalence;
- ambiguity or missing information in a supplied `normalized_statement` prevents verification of an assumption, scope restriction, quantifier, or conclusion.

This is a narrow boundary category. Do not use `possible_equivalent` merely because the mathematics is difficult, the notation is unfamiliar, the title matches, or you are generally uncertain. If a semantic difference can be identified, return `not_equivalent`.

`not_equivalent`
Return `not_equivalent` when any representative `normalized_statement` is not strictly equivalent to the current `normalized_statement`. This includes:

- related concepts or results with similar terminology;
- a mathematical definition versus a concrete application instance that merely uses that definition;
- a statement asserting existence versus a statement asserting unique existence;
- a bundled statement containing multiple conclusions or definitions versus a statement matching only some, but not all, of those conclusions or definitions;
- special versus general cases;
- stronger versus weaker assumptions or conclusions;
- different ambient settings, base objects, dimensions, regularity conditions, finiteness conditions, or parameter ranges;
- implication versus biconditional, theorem versus converse, or result versus corollary;
- incompatible definition conventions;
- omission or addition of a condition that changes the mathematical meaning;
- agreement with some, but not all, representative statements.

Output
Return exactly one valid JSON object and no Markdown fences, reasoning, scores, or additional keys:

```
{
  "relation": "equivalent"
}
```

The actual value of `relation` must be exactly one of:

- `equivalent`
- `possible_equivalent`
- `not_equivalent`

Final checks

- Use only `normalized_statement` for mathematical comparison.
- Compare the current node separately with every representative.

Prompt Template for Query-to-Candidate SEV (continued)

- Any clear mismatch forces `not_equivalent`.
- `possible_equivalent` requires genuine ambiguity in the supplied content, not model uncertainty.
- Return JSON only.

H.3 Binary SEV: Query-to-Candidate Verification

This template replaces the three-way query-to-candidate SEV prompt in the Binary SEV ablation (Appendix F.5). It retains the strict equivalence criterion while restricting the output to `equivalent` or `not_equivalent`.

Prompt Template for Binary SEV: Query-to-Candidate Verification

Role
You are an expert research mathematician responsible for strict semantic alignment in a cross-document mathematical knowledge graph. Your decisions must be conservative: merge nodes only when their complete mathematical content is strictly equivalent.

Input
You receive exactly one JSON object with the following structure:

```
{
  "current_evidence_node": {
    "evidence_node_id": "integer",
    "type": "definition | theorem",
    "title": "string",
    "field": "string",
    "subfield": "string",
    "normalized_statement": "string"
  },
  "candidate_canonical_node": {
    "canonical_node_id": "integer",
    "type": "definition | theorem",
    "title": "string",
    "field": "string",
    "subfield": "string",
    "representative_evidence_nodes": [
      {
        "role": "center_evidence | diverse_evidence_1 |
        diverse_evidence_2 | query_matched_evidence",
        "evidence_node_id": "integer",
        "normalized_statement": "string"
      }
    ]
  }
}
```

`representative_evidence_nodes` contains one to four distinct representative Evidence Nodes:

- exactly one center Evidence Node;
- at most two diverse Evidence Nodes;
- optionally one Evidence Node selected as most similar to the current node.

Some optional roles may be absent or may refer to content already represented by another role and therefore be omitted.
Use only `normalized_statement` as the mathematical content for semantic comparison. `type`, `title`, `field`, and `subfield` provide context but do not establish equivalence. The current node and candidate have already passed the same-type filter.

Prompt Template for Binary SEV: Query-to-Candidate Verification (continued)

Task
Determine whether the current Evidence Node belongs to the strict-equivalence class represented by the candidate Canonical Node.

1. Compare `current_evidence_node.normalized_statement` independently with the `normalized_statement` of every item in `candidate_canonical_node.representative_evidence_nodes`.
2. For each comparison, check the mathematical object or proposition, assumptions, ambient setting, scope, quantifiers, conclusions, and logical direction.
3. Aggregate the individual comparisons using the relation rules below.

Do not combine different representative statements into a new statement. Do not use majority voting. The current Evidence Node is strictly equivalent to the Canonical Node only when it is strictly equivalent to every supplied representative statement.

Relation rules

`equivalent`
Return `equivalent` if and only if the current `normalized_statement` is strictly equivalent to every representative `normalized_statement`:

- the same mathematical object or construction is defined, or the same proposition is asserted;
- assumptions, ambient setting, scope, quantifiers, conclusions, and logical direction agree;
- differences are limited to notation, variable names, order of presentation, standard definitional expansion, or logically equivalent reformulation.

Textual identity is not required. Mathematical equivalence under exactly the same assumptions is required.

`not_equivalent`
Return `not_equivalent` when any representative `normalized_statement` is not strictly equivalent to the current `normalized_statement`. This includes:

- related concepts or results with similar terminology;
- a mathematical definition versus a concrete application instance that merely uses that definition;
- a statement asserting existence versus a statement asserting unique existence;
- a bundled statement containing multiple conclusions or definitions versus a statement matching only some, but not all, of those conclusions or definitions;
- special versus general cases;
- stronger versus weaker assumptions or conclusions;
- different ambient settings, base objects, dimensions, regularity conditions, finiteness conditions, or parameter ranges;
- implication versus biconditional, theorem versus converse, or result versus corollary;
- incompatible definition conventions;
- omission or addition of a condition that changes the mathematical meaning;
- agreement with some, but not all, representative statements.

Output
Return exactly one valid JSON object and no Markdown fences, reasoning, scores, or additional keys:

```
{
  "relation": "equivalent"
}
```

Prompt Template for Binary SEV: Query-to-Candidate Verification (continued)

The actual value of `relation` must be exactly one of:

- `equivalent`
- `not_equivalent`

Final checks

- Use only `normalized_statement` for mathematical comparison.
- Compare the current node separately with every representative.
- Any clear mismatch forces `not_equivalent`.
- Return JSON only.

H.4 Claude Code Agent

The Claude Code Agent baseline uses this template to assess retrieved canonical candidates, request up to two additional searches, and return a unique canonical target or `NIL` (Appendix C).

Prompt Template for Claude Code Agent

Role
You are an autonomous retrieval agent for open-world mathematical semantic equivalence resolution in algebra. Your decisions must be conservative: merge only when strict mathematical equivalence can be established.

Objective
You receive one query Evidence Node and a retrieval result from a frozen Canonical Knowledge Graph. Determine whether the query is strictly mathematically equivalent to exactly one retrieved Canonical Node.
Return that Canonical Node ID when a unique strictly equivalent target exists. Return NIL when no strictly equivalent target exists or when a unique target cannot be established.

Input
The user supplies exactly one JSON object:

```
{
  "query_evidence_node": {
    "query_evidence_node_id": "integer",
    "type": "definition | theorem",
    "title": "string",
    "field": "string",
    "subfield": "string",
    "normalized_statement": "string"
  },
  "retrieval_result": {
    "query_evidence_node_id": "integer",
    "search_id": "integer",
    "search_type": "original | reformulated",
    "search_query": "string",
    "ranking_source": "embedding | reranker",
    "candidate_count": "integer",
    "candidates": [
      {
        "rank": "integer",
        "canonical_node_id": "integer",
        "type": "definition | theorem",
        "title": "string",
        "representative_evidence_nodes": [
          {
```

Prompt Template for Claude Code Agent (continued)

```
            "role": "center_evidence | diverse_evidence_1 |
            diverse_evidence_2 | query_matched_evidence",
            "normalized_statement": "string"
          }
        ]
      }
    ]
  }
}
```

Each retrieval result contains at most 20 Canonical candidates. Additional retrieval results returned by the tool use the same structure.

Mathematical equivalence
Use only `normalized_statement` as the authoritative mathematical content. Titles, fields, subfields, and retrieval ranks are supporting information only and do not establish equivalence.
Strict equivalence requires the same mathematical definition or proposition, including compatible mathematical objects, assumptions, ambient setting, scope, quantifiers, conclusions, and logical direction. Differences may be limited to notation, variable names, order of presentation, standard definitional expansion, or logically equivalent reformulation.
Topical similarity, shared terminology, a special or general case, a stronger or weaker statement, a converse, a corollary, or partial overlap is not strict equivalence.
A mathematical definition is not equivalent to a concrete application instance that merely uses that definition. Existence is not equivalent to unique existence. A bundled statement containing several definitions or conclusions is not equivalent to a statement matching only part of that bundle.
A query belongs to a Canonical Node only if it is strictly compatible with every representative Evidence statement supplied for that node. A clear conflict with any representative statement prevents merging.

Retrieval workflow
The initial retrieval result was produced automatically from the original query Evidence Node using the fixed Gangmu Evidence-level retrieval pipeline. You may analyze every supplied candidate freely.
If the available results are insufficient, you may call `search_again` with a reformulated standalone mathematical statement. The reformulation must preserve the complete mathematical meaning and type of the original query. Do not drop assumptions, quantifiers, conclusions, or bundled components merely to improve retrieval similarity.
The original search counts as the first search. You may request at most two additional searches. The tool enforces this limit. Use another search only when it could discover a missing strict equivalent or resolve ambiguity.

Decision process
After each retrieval result, choose one action:

1. Return a merge prediction if exactly one retrieved Canonical Node can be established as strictly equivalent to the complete query.
2. Return NIL if no retrieved candidate is strictly equivalent.
3. Call `search_again` if the available information is insufficient and an additional search remains available.
4. Return NIL if the search budget is exhausted and a unique strictly equivalent target still cannot be established.

Do not output pairwise relation labels or intermediate candidate classifications. Do not assume the highest-ranked candidate is equivalent. If several candidates remain plausible and no unique target can be established, return NIL.

Restrictions

- Do not modify retrieval results.

Prompt Template for Claude Code Agent (continued)

- Do not select a Canonical Node that did not appear in a retrieval result.
- Do not access external sources, files, web pages, or information outside the supplied query and retrieval results.
- Do not create or modify Canonical Nodes.
- Do not ask the user for help.
- Treat API and tool failures as errors, not as evidence for NIL.

Final output
Return exactly one JSON object and no Markdown fences or additional text:

```
{
  "prediction": "merge | nil",
  "canonical_node_id": "integer | null",
  "brief_reason": "string"
}
```

When `prediction` is `merge`, `canonical_node_id` must identify exactly one Canonical Node that appeared in a retrieval result. When `prediction` is `nil`, `canonical_node_id` must be null.
`brief_reason` must concisely state the decisive mathematical agreement or difference. Do not provide hidden chain-of-thought or a long derivation.