RICH-NODE GRAPHS FOR CORPUS-GROUNDED LONG-FORM SFT DATA SYNTHESIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Domain-specific supervised fine-tuning (SFT) requires training data that cover related subtopics while remaining grounded in source evidence. Existing pipelines often rely on parametric knowledge, individual passages, or flat retrieved contexts, limiting their ability to organize distributed evidence into coherent supervision. We present **RiNoGen**, a framework that transforms domain corpora into mixed-length question–answer pairs using source-retaining Rich-Node graphs. RiNoGen selects corpus-supported seed entities, expands each seed with retrieved evidence, and applies Map–Reduce synthesis to integrate local subtopics into a long-form answer. Qwen3-8B serves as both the answer generator and the model fine-tuned on the resulting data, so answer generation does not require a larger teacher. On a corpus of 670k Chinese educational papers, RiNoGen constructs 30,000 SFT examples. In a complete-pipeline comparison under a fixed 3,000-example output-data budget, it obtains 87.59% closed-book accuracy on a fixed 427-item evaluation subset, 1.14 percentage points above the strongest compared baseline, although upstream corpus access is not matched. Internal controls and ablations support hierarchical evidence organization and local-to-global synthesis. On 99 selected time-sensitive topics from 2025–2026, an LLM judge prefers the optional RiNoGen-Web branch to its local-only counterpart in 89 cases.

1 INTRODUCTION

Domain-specific supervised fine-tuning (SFT) data construction turns document collections into instruction–response pairs that transfer corpus knowledge to a model. It is particularly important for education, where knowledge is spread across large paper collections and manual annotation is costly. For broad topics, useful supervision must simultaneously retain traceable evidence, cover related subtopics, and synthesize them into a coherent long-form answer. This is difficult because source corpora are organized as documents and passages, whereas training examples are organized around answerable topics.

Current automated methods address only part of this transformation. Self-instruction and distillation scale example generation Wang et al. (2023); Xu et al. (2024); Taori et al. (2023); Mukherjee et al. (2023), but their coverage and temporal currency depend on model knowledge. Passage-grounded generation supplies evidence but usually captures one local aspect of a topic. RAG retrieves a broader top-k context Lewis et al. (2020), while graph-based retrieval organizes distributed evidence for query answering Edge et al. (2024). These systems are generally designed to answer an input query, rather than to expand a corpus-supported topic and repeatedly convert its local and global evidence into SFT examples. The missing step is therefore not retrieval alone, but an intermediate representation that records where each subtopic came from and how it relates to the topic being expanded.

We address this gap with **RiNoGen**. After corpus vectorization, the framework proceeds in three stages: seed entity generation, Rich-Node graph construction, and Map–Reduce QA synthesis. Seed entities are selected through clustering and extensibility filtering, then expanded into hierarchical **Rich-Node** (RN) graphs. Each RN records an entity-centered summary, expansion clues, retrieved sources, graph depth, and domain information. Map generation produces subgraph-level sections, and Reduce synthesis integrates them into a mixed-length answer before regenerating the question.

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.

The main experiments use local retrieval; RiNoGen-Web supplies web evidence through the same node interface for the time-sensitive evaluation. The contributions are:

- We introduce provenance-aware Rich-Node graphs with seed-driven expansion and Map–Reduce synthesis for constructing mixed-length SFT examples; controlled variants assess the complete graph, local Map, and global Reduce components.
- From a commercially acquired corpus of 670k Chinese educational papers, we construct 30,000 retrieval-conditioned, length-filtered, and near-deduplicated SFT examples, together with a 10,000-item test pool whose fixed 427-item subset is used in the reported educational evaluation.
- In a complete-pipeline comparison with 3,000 output examples per method but method-specific upstream corpus access, RiNoGen obtains 87.59% accuracy, 1.14 points above the strongest compared baseline. An LLM judge also prefers RiNoGen-Web in 89 of 99 paired comparisons with RiNoGen local-only on selected 2025–2026 topics.

Within the evaluated Chinese educational QA setting, these results support the premise that organizing retrieved evidence before synthesis can improve the utility of domain-specific SFT data under a fixed output-example budget.

2 Related Work

2.1 Synthetic Data Generation for Supervised Fine-Tuning

Synthetic instruction generation derives SFT examples from seed tasks or teacher-model outputs. Self-Instruct Wang et al. (2023) bootstraps instructions from a seed set, while Evol-Instruct Xu et al. (2024) iteratively rewrites instructions to increase their complexity. Alpaca Taori et al. (2023) distills instruction-following behavior, and Orca Mukherjee et al. (2023) learns from explanation traces produced by stronger models. These methods scale instruction diversity and behavioral supervision, but they do not by themselves organize evidence distributed across a domain corpus. Corpus-grounded long-form SFT additionally requires coordinating multiple passages while preserving their provenance.

GraphGen Chen et al. (2025) constructs a fine-grained knowledge graph from source text, identifies trainee-model knowledge gaps using expected calibration error, and uses multi-hop neighborhood sampling to generate atomic, aggregated, and multi-hop QA pairs. RiNoGen instead expands retrieval-grounded seeds into hierarchical Rich-Node graphs and applies Map–Reduce synthesis to subgraph-level and global evidence. The methods therefore differ in their data-selection signal, graph representation, and target output: GraphGen targets model knowledge gaps, whereas RiNoGen targets provenance-aware long-form SFT construction.

The external comparison also includes prompt-based corpus transformation methods—WRAP Maini et al. (2024), Genie Yehudai et al. (2024), LongForm Köksal et al. (2024), and SELF-QA Zhang & Yang (2023)—and the graph-based EntiGraph procedure Yang et al. (2025). They cover direct or reverse-instruction transformation, question generation followed by answering, and entity-based expansion. RiNoGen instead expands retrieved evidence around each seed before producing an integrated answer, motivating a complete-pipeline comparison under a common output count.

2.2 Retrieval-Augmented Generation

Retrieval-augmented generation (RAG) conditions a language model on passages selected from an external collection Lewis et al. (2020). Dense Passage Retrieval Karpukhin et al. (2020) uses bi-encoder representations for semantic matching, while BGE-M3 Chen et al. (2024) supports multilingual retrieval and inputs ranging from short sentences to long documents. These retrieval methods identify relevant passages but do not themselves represent the subtopic relations and expansion paths needed to construct multi-aspect SFT examples. RiNoGen therefore uses dense retrieval as an evidence layer and adds hierarchical organization before synthesis.

2.3 Graph-Based Retrieval and Knowledge Organization

Graph-based RAG makes relationships among retrieved evidence explicit. GraphRAG Edge et al. (2024) builds an entity graph, detects communities, and summarizes them for global questions. LightRAG Guo et al. (2025) uses graph and vector indexes for dual-level retrieval of entities and relations. HippoRAG Gutiérrez et al. (2024) constructs a knowledge graph and applies personalized PageRank for multi-hop retrieval.

These systems use graphs primarily to answer input queries over an indexed corpus. RiNoGen instead uses a graph as an intermediate representation for training-data construction: each Rich-Node retains retrieved evidence, expansion clues, and source records, while Map–Reduce synthesis converts local and global graph content into mixed-length SFT examples. The graph is therefore not only a retrieval index: it determines which subtopics are expanded, which evidence is grouped locally, and which local sections enter final synthesis.

3 Proposed Method

Figure 1 outlines RiNoGen. Corpus vectorization provides a shared index for seed selection and retrieval. The pipeline then selects seed entities, expands each seed into a **Rich-Node** (RN) graph, and synthesizes QA pairs with a Map–Reduce procedure. An RN stores an entity summary, expansion clues, retrieved sources, graph depth, and domain information. Retrieval supplies node context; graph expansion and Map–Reduce determine its organization. Main experiments use local evidence, while RiNoGen-Web is evaluated separately. Qwen3-8B serves as both answer generator and fine-tuning target.

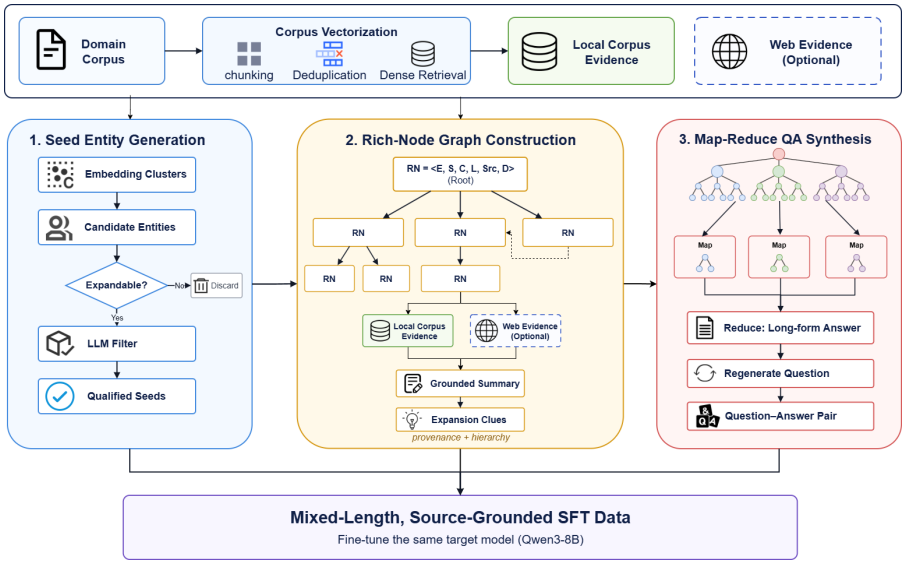

Figure 1: Overview of the RiNoGen data construction pipeline.

3.1 Corpus Vectorization

We split each document into sentence-preserving chunks. For Chinese text, the segmenter detects full stops, exclamation marks, question marks, and newline boundaries, and appends complete sentences until a chunk reaches the default limit of 500 characters, avoiding arbitrary within-sentence boundaries.

For incremental ingestion, we maintain an MD5 fingerprint cache. Each fingerprint is computed from the document source identifier and its first 200 characters, and previously observed documents are skipped. We encode the remaining chunks with BGE-M3 and store the vectors in a flat FAISS index. The same vectors are exported as `.npy` tensors for seed clustering, so seed discovery and node grounding use one embedding space.

3.2 SEED ENTITY GENERATION

Graph expansion requires seeds that are both represented in the corpus and broad enough to yield multiple subtopics. We obtain these seeds through clustering, entity extraction, and extensibility filtering.

We first apply MiniBatch K-Means to the chunk embeddings. The reported experiments use $K_{\text{cluster}} = 5{,}000$, selected by the hyperparameter analysis. We discard clusters containing fewer than 50 chunks and retain the 50 chunks closest to each remaining centroid. Part-of-speech tagging extracts terms in the noun categories `n`, `nr`, `ns`, `nt`, and `nz`. We remove high-frequency generic terms, numeric strings, single-character tokens, and duplicates, then select one candidate entity per cluster by frequency vote.

Frequency alone does not indicate whether a candidate can support multi-level expansion. We therefore ask Qwen3-8B to score, on a 0–10 scale, whether each candidate can yield at least five meaningful subtopics. Candidates scoring below 7 are removed; failed API calls are retried up to three times. The retained candidates initialize the graph roots. This second filter removes frequent but narrow terms that do not match the targeted multi-aspect answers.

3.3 RICH-NODE GRAPH CONSTRUCTION

The RN graph records both the evidence retrieved for an entity and the path by which that entity was reached. We represent it as a directed graph: nodes contain entity-level evidence, and edges encode expansion relations between entities.

Each Rich-Node is a six-tuple:

$$RN = \langle E,\, S,\, C,\, L,\, Src,\, D \rangle, \tag{1}$$

with fields defined in Table 1. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the RN graph. Selecting an entity from the clue set of node v_i adds an edge

$$e_{ij} = (v_i, v_j, t_{ij}), \quad t_{ij} \in \{\text{EXPAND}, \text{REFERENCE}\}. \tag{2}$$

An EXPAND edge creates child node v_j; a REFERENCE edge links to an existing node when another branch reaches the same entity.

Table 1: Fields of the Rich-Node six-tuple.

Symbol	Field	Description
E	Entity	Entity name, for example *Deep Learning*
S	Summary	Entity summary conditioned on retrieved evidence
C	Clues	Expansion pointer set, $C = \{c_1, \ldots, c_{k_c}\}$, $k_c \in [3, 5]$
L	Layer	Depth assigned during layer-wise expansion
Src	Sources	Retrieved evidence used during node construction
D	Domain	Macro domain used for domain-specific prompting

For an entity E, we retrieve the top-k_r chunks from the FAISS index and condition the generator on these chunks to produce summary S and clue set C. The main experiments retrieve only from the local corpus. When web retrieval is enabled, its passages enter the same evidence pool before summary and clue generation. Nodes with insufficient evidence are marked as low confidence for later exclusion or manual inspection.

We expand each seed layer by layer. The domain field D is predicted once per root and cached for its descendants; it selects the prompt template but is not part of the supervision target. At each node, the model summarizes the retrieved evidence and extracts $k_c \in [3, 5]$ clue entities from that summary.

For candidate clue c, we require $s(c, E_{\text{root}}) > 0.3$ and $s(c, E_{\text{parent}}) \in [0.4, 0.85]$, where $s(\cdot, \cdot)$ denotes cosine similarity. The root constraint preserves topic relevance; the parent interval requires a meaningful local relation while filtering near duplicates. An accepted clue creates an EXPAND edge if the entity is new and a REFERENCE edge otherwise. Expansion then proceeds to the next layer. Reusing an existing node through a REFERENCE edge preserves cross-branch connections without duplicating the node's retrieved evidence and summary.

3.4 MAP–REDUCE QA SYNTHESIS

The **Map** stage groups graph content into local subgraphs rooted at first-layer nodes. For each subgraph, the generator conditions on the graph-root summary and the corresponding descendant summaries to produce one answer section, keeping generation focused on one subtopic.

The **Reduce** stage integrates the local sections into a long-form answer and resolves their ordering and overlap. The question is regenerated from the final answer to target the retained content. We request one of three answer lengths: 500–1,000, 1,000–2,000, or 2,000–5,000 Chinese characters. We discard outputs shorter than the requested minimum and remove near-duplicate QA pairs by cosine similarity. The remaining length-filtered and near-deduplicated pairs constitute the SFT corpus.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We first characterize the generated corpora, then compare their closed-book training utility under fixed output-example budgets. Internal controls contrast parametric generation, passage grounding, flat retrieval, and structured graph synthesis; external baselines provide a complete-pipeline comparison. We then examine components, hyperparameters, and RiNoGen-Web. Because answer lengths differ, equal example counts do not imply equal token or compute budgets.

Training Protocol All SFT variants fine-tune Qwen3-8B Qwen Team (2025) with the same LoRA configuration. We train for two epochs in bf16 with a maximum sequence length of 8,192, a per-device batch size of 1, and 16 gradient-accumulation steps. AdamW uses a learning rate of 5×10^{-5}, cosine decay, a 0.03 warmup ratio, and 0.01 weight decay. LoRA uses rank 16, $\alpha = 32$, and 0.01 dropout on the attention, MLP, and output projection layers. Training runs on four GPUs with DeepSpeed ZeRO-2 and gradient checkpointing.

Data and Evaluation Budgets RiNoGen produces 30,000 examples from 670k educational papers, divided equally among the three ranges defined in Map–Reduce QA Synthesis. The internal-control study uses 3,600 examples per fine-tuned variant; the external comparison uses 3,000 per method. We also construct a 10,000-item educational multiple-choice pool. All reported educational accuracy results use the same fixed random subset of 427 items. Fixing the evaluation subset, backbone, and LoRA protocol controls these factors across methods; the protocols below specify their upstream evidence.

Pre-training Diagnostics Before SFT, we characterize generated answers using compression ratio (compressed/original bytes), unique character 2-grams and 3-grams, type-token ratio, and entity density. These length-sensitive proxies describe redundancy and lexical coverage; downstream accuracy remains the primary measure of training utility. Definitions appear in Appendix C.

Evaluation Benchmark We sample source passages for the test pool from the same paper corpus, with at most 10 chunks per file and at least 150 characters per chunk. Qwen3-32B generates four-option, single-answer questions that test transferable domain principles rather than paper-specific findings. Each question must be self-contained. Prompted completeness and correctness filters reject passage references, verify the labeled answer, and check that all distractors can be eliminated. We evaluate each method 20 times at temperature 0 and report mean top-1 accuracy; the variance is below 0.1 for every method.

Training–Evaluation Leakage We compare the serialized Ours-Train split with all 427 evaluation items using normalized character 5-gram overlap and BGE-M3 cosine similarity. No exact containment is found. The maximum containment, Jaccard, and cosine scores are 0.2339, 0.00755, and 0.8091, respectively; all items remain below the predefined medium-risk thresholds. This audit covers the evaluated subsets, not the complete 30,000-example resource and 10,000-item test pool.

4.2 Comparison with Internal Controls

We compare four construction strategies under a 3,600-example budget. **Cut** segments the corpus with a 500-character window and 50-character overlap, matches chunks to seeds with Aho–Corasick search, and generates one QA pair per matched chunk, capped at five chunks per seed. **NoExpanded** regenerates answers to the same questions from Qwen3-8B's parametric knowledge. **NoExpanded+RAG** prepends the top-5 FAISS passages but omits hierarchical expansion. **RiNoGen (Ours)** grounds node summaries in local retrieval and synthesizes each answer from a multi-layer RN graph using Map–Reduce.

Table 2: Pre-training corpus diagnostics of different data construction methods.

Metric	Cut	NoExpanded	NoExpanded+RAG	**RiNoGen (Ours)**
Compression Ratio ↑	0.4526	0.4461	0.4787	**0.5276**
Unique 2-gram ↑	0.7553	0.7127	0.8050	**0.8926**
Unique 3-gram ↑	0.8970	0.8559	0.9342	**0.9760**
TTR ↑	0.3541	0.3256	0.3747	**0.4644**
Entity Density ↑	0.3096	0.2880	0.3100	**0.3526**
Answer grounding	single-chunk	none	flat retrieval	multi-layer graph
Answer length	short	short	configurable	configurable

RiNoGen has the highest value on each reported diagnostic in Table 2. Under the definitions in Table 7, this corresponds to lower measured redundancy and higher phrase diversity, lexical richness, and entity density. Because the methods differ in answer-length configuration and the proxies are length-sensitive, the table characterizes the corpora rather than factuality or the isolated effect of graph structure.

Training Variants. We fine-tune one Qwen3-8B model on 3,600 examples from each strategy using the fixed protocol in the experimental setup. **Base** is the untuned model; **Base+RAG** adds retrieved passages only at inference. All fine-tuned variants are evaluated closed-book.

Table 3: Internal-control accuracy (%) on the fixed 427-item subset, averaged over 20 runs at temperature 0. All variances are below 0.1. †: inference-time retrieval without fine-tuning.

Method	#Samples	Acc. (%)
Base (Qwen3-8B)	—	83.4
Base + RAG†	—	83.8
Cut-Train	3,600	85.6
NoExpanded-Train	3,600	84.0
NoExpanded+RAG-Train	3,600	83.6
Ours-Train	3,600	**86.4**

Table 3 shows that inference-time RAG yields a 0.4-point gain over Base (83.8% vs. 83.4%). Among the fine-tuned controls, NoExpanded-Train and NoExpanded+RAG-Train reach 84.0% and 83.6%, respectively, whereas passage-grounded Cut-Train reaches 85.6%. Ours-Train obtains the highest observed score, 86.4%, improving over Base by 3.0 points and Cut-Train by 0.8 points. The Cut result indicates that converting retrieved passages into training examples is more effective here than relying only on parametric generation. Flat retrieval does not improve NoExpanded, whereas the 0.8-point Ours-Train gain over Cut-Train is consistent with a benefit from evidence organization. This margin remains modest on 427 items, and conditions are matched by examples rather than tokens.

4.3 Comparison with External Data-Construction Methods

Comparison Protocol. We compare RiNoGen with GraphGen Chen et al. (2025), EntiGraph Yang et al. (2025), Genie Yehudai et al. (2024), LongForm Köksal et al. (2024), SELF-QA Zhang & Yang (2023), and WRAP Maini et al. (2024). The six external methods receive the same 7,000-chunk input slice, sampled with seed 42 from the educational corpus; RiNoGen retains its native retrieval-backed construction process. Each method supplies exactly 3,000 SFT examples for the fixed Qwen3-8B LoRA protocol. All resulting models are evaluated closed-book on the same 427-item benchmark. The experiment therefore fixes output count and downstream training, but not upstream corpus access or construction cost.

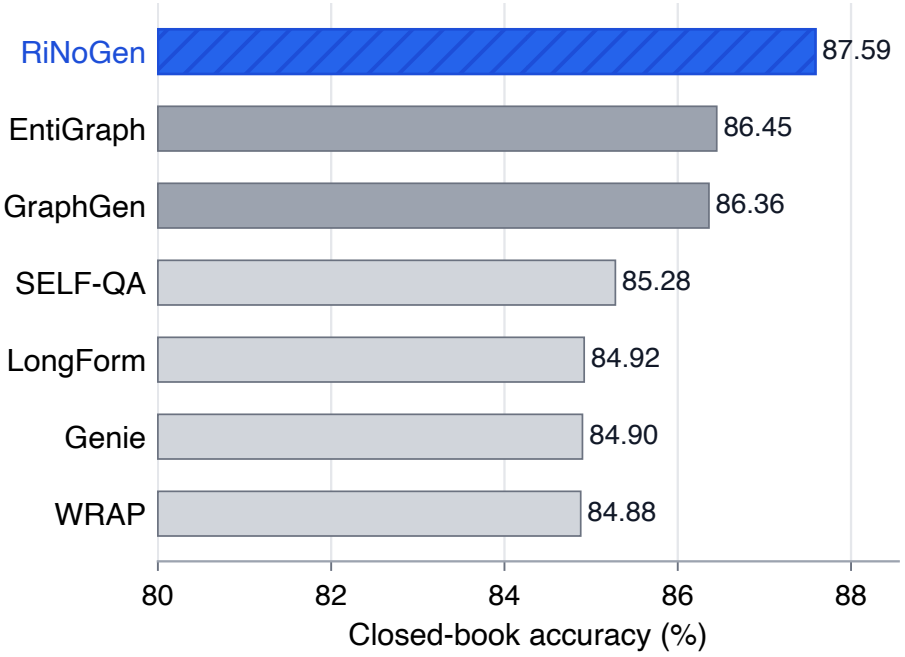

Figure 2: Closed-book accuracy on the fixed 427-item subset. External methods share a 7,000-chunk input slice, and every method supplies 3,000 SFT examples. The horizontal axis begins at 80% for readability.

Figure 2 shows an observed accuracy of 87.59% for RiNoGen, 1.14 points above EntiGraph, 1.23 points above GraphGen, and 2.31–2.71 points above the remaining methods. This comparison measures the utility of the examples produced by each complete pipeline under the same output count and training protocol. It does not isolate the Rich-Node representation because external methods share a fixed input slice, while RiNoGen uses its native index. The gain also shows that same-model, retrieval-conditioned adaptation can improve the downstream score, but does not distinguish knowledge acquisition from format or style adaptation.

4.4 Ablation Study

We assess the contribution of the Rich-Node generation stages by constructing three ablated variants from the same saved procedural records. `root_only` keeps only the seed-level root Rich-Node and generates the final answer directly from that node. `map_only` builds the complete Rich-Node graph, generates local subgraph-level sections, and concatenates those sections directly. `reduce_only` builds the complete Rich-Node graph, omits subgraph-level Map generation, and feeds all node summaries into one synthesis prompt. `root_only` is composite: it removes expansion and replaces Map–Reduce with direct generation.

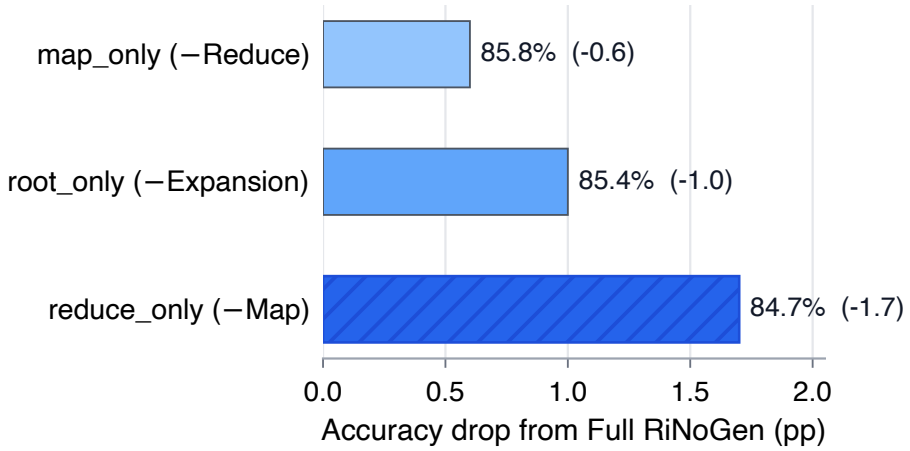

Figure 3: Accuracy drops relative to Full RiNoGen (86.4%) on the fixed 427-item subset. Labels report absolute accuracy and the drop in percentage points.

Figure 3 shows that the variant without subgraph-level Map generation has the largest drop, from 86.4% to 84.7% (-1.7 points). The composite `root_only` variant is 1.0 point below Full RiNoGen, while omitting global Reduce synthesis costs 0.6 points. All variants remain above the Base result of 83.4%, and the complete pipeline has the highest observed accuracy. The results support distinct roles for Map and Reduce; the `root_only` drop cannot be attributed to expansion alone.

4.5 Configuration Selection

Conditional diagnostic sweeps over answer length, seed cluster count, and retrieved-passage count motivate an equal mix of the three answer-length ranges, 5k seed clusters, and top-$k_r = 5$ retrieval. These choices were made from pre-training diagnostics rather than jointly optimized downstream accuracy; full results appear in Appendix D.

4.6 Evaluation on Time-Sensitive Web Knowledge

We next evaluate the optional RiNoGen-Web branch on selected time-sensitive topics. We construct 100 seed topics concerning publicly searchable events and developments from 2025–2026. Four Qwen3-8B-based configurations generate an answer for each seed: Base Qwen3-8B, Naive Web RAG, RiNoGen local-only, and RiNoGen-Web. After matching records by seed, 99 topics are shared across all four groups. An LLM judge performs pairwise comparison with candidate order randomized. It evaluates temporal accuracy, factual correctness, evidence grounding, coverage depth, and coherence on a five-point scale. For comparisons involving retrieved web content, the judge receives an answer-relevant and length-controlled evidence set, with at most five passages selected per candidate. We report pairwise preferences and two-sided sign tests over non-tied judgments; one failed RiNoGen-Web versus Base judgment is excluded.

Effectiveness of Web Retrieval. Table 4 shows that the LLM judge consistently prefers configurations with access to current web content. RiNoGen-Web wins 89 of 99 comparisons against RiNoGen local-only, corresponding to an 89.9% win rate ($p < 0.001$). Naive Web RAG similarly wins 88 of 99 comparisons against Base Qwen3-8B (88.9%, $p < 0.001$). When compared directly with Base Qwen3-8B, RiNoGen-Web wins 66 of 98 valid comparisons (67.3%, $p = 0.00077$). The RiNoGen-Web versus RiNoGen local-only comparison changes access to web evidence within the same pipeline. Under automatic judging, the result supports a benefit on these selected topics, but neither ranks model architectures nor replaces human factuality assessment.

Table 4: LLM-judged pairwise evaluation on time-sensitive 2025–2026 topics. Win–loss counts and win rates refer to the first-listed method. Sign-test p-values are computed over non-tied judgments.

Comparison	Win–loss	Win rate (%)	p
RiNoGen-Web vs. RiNoGen local-only	89–10	89.9	< 0.001
Naive Web RAG vs. Base Qwen3-8B	88–11	88.9	< 0.001
RiNoGen-Web vs. Base Qwen3-8B	66–32	67.3	0.00077

4.7 Discussion

Internal controls and ablations support the value of structured expansion and local-to-global synthesis; removing Map produces the largest drop. The evidence comes from one model family and a 427-item subset. Upstream corpus access and token budgets differ across methods, the leakage audit cannot exclude source-document overlap, and web results rely on LLM judgment. Token-matched, document-disjoint, human-evaluated studies remain necessary.

5 Conclusion

RiNoGen organizes retrieved evidence into source-retaining Rich-Node graphs and synthesizes mixed-length SFT examples through Map–Reduce generation. Using Qwen3-8B as both generator and training target, it produced 30,000 examples from 670k Chinese educational papers. Under a fixed 3,000-example output budget, the complete pipeline reached 87.59% accuracy on the 427-item subset, 1.14 points above the strongest compared baseline. Internal controls and ablations support structured expansion and local-to-global synthesis; on selected 2025–2026 topics, an LLM judge preferred RiNoGen-Web to its local-only counterpart in 89 of 99 comparisons. These findings motivate further evaluation across domains, backbones, and human factuality judgments.

AI use statement

Large language models were used as part of the research method to generate candidate supervised fine-tuning data, generate candidate evaluation items, and provide pairwise judgments in the time-sensitive web evaluation, as described in the paper. AI-assisted tools also supported experiment design, code development, manuscript editing, and conversion to the ICLR format. The authors take responsibility for the experimental results, citations, and final manuscript.

Reproducibility statement

The paper describes the data construction stages and the training and evaluation protocols in Sections 3 and 4. The appendix reports the training–evaluation similarity audit and additional benchmark results. The underlying paper corpus is commercially acquired, which may limit redistribution of source documents.

References

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2318–2335. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.137.

Zihong Chen, Wanli Jiang, Jinzhe Li, Zhonghang Yuan, Huanjun Kong, Wanli Ouyang, and Nanqing Dong. Graphgen: Enhancing supervised fine-tuning for llms with knowledge-driven synthetic data generation, 2025. URL https://arxiv.org/abs/2505.20416.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization, 2024. URL https://arxiv.org/abs/2404.16130.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile Van Krieken, and Pasquale Minervini. Are we done with MMLU? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5069–5096, 2025. doi: 10.18653/v1/2025.naacl-long.262. URL https://aclanthology.org/2025.naacl-long.262/.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 10746–10761. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.568.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.14831.

Yun He, Di Jin, Chaoqi Wang, Chloe Bi, Karishma Mandyam, Hejia Zhang, Chen Zhu, Ning Li, Tengyu Xu, Hongjiang Lv, Shruti Bhosale, Chenguang Zhu, Karthik Abinav Sankararaman, Eryk Helenowski, Melanie Kambadur, Aditya Tayade, Hao Ma, Han Fang, and Sinong Wang. Multi-IF: Benchmarking LLMs on multi-turn and multilingual instructions following. *arXiv preprint arXiv:2410.15553*, 2024. URL https://arxiv.org/abs/2410.15553.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.550.

Abdullatif Köksal, Timo Schick, Anna Korhonen, and Hinrich Schütze. Longform: Effective instruction tuning with reverse instructions. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.414.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119. Association for Computational Linguistics, 2016. doi: 10.18653/v1/N16-1014.

Pratyush Maini, Skyler Seto, Richard H. Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. Rephrasing the web: A recipe for compute and data-efficient language modeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14044–14072. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.757.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4, 2023. URL https://arxiv.org/abs/2306.02707.

Shishir G. Patil, Huanzhi Mao, Charlie Cheng-Jie Ji, Fanjia Yan, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://gorilla.cs.berkeley.edu/leaderboard.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim

Rocktäschel, and Sebastian Riedel. Kilt: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2523–2544. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.200.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Junyi Sun. Jieba chinese text segmentation. Software, 2012. URL https://github.com/fxsjy/jieba.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpaca: A strong, replicable instruction-following model. Stanford Center for Research on Foundation Models, 2023. URL https://crfm.stanford.edu/2023/03/13/alpaca.html.

Mildred C. Templin. *Certain Language Skills in Children: Their Development and Interrelationships*. University of Minnesota Press, Minneapolis, MN, 1957.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13484–13508. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.754.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/82eec786fdfbbfa53450c5feb7d1ac92-Abstract-Conference.html.

Zitong Yang, Neil Band, Shuangping Li, Emmanuel Candès, and Tatsunori Hashimoto. Synthetic continued pretraining. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=07yvxWDSla.

Asaf Yehudai, Boaz Carmeli, Yosi Mass, Ofir Arviv, Nathaniel Mills, Assaf Toledo, Eyal Shnarch, and Leshem Choshen. Genie: Achieving human parity in content-grounded datasets generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=RjYKTQ0L0W.

Xuanyu Zhang and Qing Yang. Self-qa: Unsupervised knowledge guided language model alignment, 2023.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023. URL https://arxiv.org/abs/2311.07911.

Jacob Ziv and Abraham Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, 1977. doi: 10.1109/TIT.1977.1055714.

A Training–Evaluation Leakage Audit

We conduct a record-level leakage audit between the serialized RiNoGen training split used for Ours-Train and the 427-item educational rapid-evaluation subset. The audit script parses 3,600 training records from the serialized training file and all 427 evaluation items.

A.1 Lexical Similarity Analysis

The lexical audit concatenates the instruction and response fields of each training example. For every evaluation item, it concatenates the question, answer label, answer options, reference, and explanation fields when available. All texts are normalized with Unicode NFKC normalization,

lowercasing, and removal of whitespace and punctuation. We then construct sets of character 5-grams.

For each evaluation item e and training example t, we compute 5-gram containment and Jaccard similarity:

$$\text{Containment}(e,t) = \frac{|G_e \cap G_t|}{|G_e|}, \tag{3}$$

$$\text{Jaccard}(e,t) = \frac{|G_e \cap G_t|}{|G_e \cup G_t|}, \tag{4}$$

where G_e and G_t denote the normalized character 5-gram sets. Each evaluation item is assigned the maximum score observed over all training examples. An item is labeled high risk when exact containment is detected, containment is at least 0.80, or Jaccard similarity is at least 0.80. Medium risk corresponds to containment of at least 0.60 or Jaccard similarity of at least 0.50. All remaining items are labeled low risk.

A.2 Embedding Similarity Analysis

We complement the lexical analysis with semantic similarity. BGE-M3 encodes the concatenated training and evaluation texts, truncated to at most 1,200 characters. The embeddings are normalized, and the audit computes the maximum cosine similarity between each evaluation item and all training examples. Cosine similarity values of at least 0.92 and 0.85 define high and medium embedding risk, respectively. The ten most similar evaluation–training pairs are exported for focused inspection.

A.3 Audit Results

Table 5 summarizes the distributions of the maximum similarity scores. The lexical audit detects no exact containment. All 427 evaluation items fall into the low-risk category under the 5-gram criteria. The maximum containment score is 0.2339, and the maximum Jaccard similarity is 0.00755. The embedding audit also assigns all 427 items to the low-risk category. The mean maximum cosine similarity is 0.6634, and the largest value is 0.8091, below the medium-risk threshold of 0.85.

Table 5: Record-level similarity between the serialized RiNoGen training split used for Ours-Train and the 427-item educational rapid-evaluation subset. P95 denotes the 95th percentile and Risk denotes medium/high-risk matches.

Metric	Mean	Med.	P95	Max	Risk
5-gram containment	0.05757	0.05172	0.11290	0.23387	0/427
5-gram Jaccard	0.00163	0.00132	0.00410	0.00755	0/427
BGE-M3 cosine	0.66336	0.66739	0.73488	0.80913	0/427

These results provide evidence against direct or near-duplicate training–evaluation overlap at the record level. The shared educational domain naturally produces topical similarity, which is reflected in the moderate embedding scores. The audit estimates lexical and semantic overlap between serialized examples. Source-document and source-chunk provenance would be required for a document-level contamination analysis. The audit applies to the experimental subsets. A full cross-comparison of all 30,000 training examples and all 10,000 test items remains future work.

B General Capability Evaluation

To examine how RiNoGen-based SFT affects capabilities beyond the target educational QA setting, we compare the base model with *base+ours_train* on four general benchmarks. MMLU-Redux Gema et al. (2025) measures **General Tasks** ability through general knowledge and reasoning. IFEval Zhou et al. (2023) evaluates **Alignment Tasks** ability through constrained instruction following. Multi-IF He et al. (2024) measures multilingual instruction following under **Multilingual Tasks**. BFCL v3 Patil et al. (2025) serves as a proxy for **Agent & Coding** ability

Table 6 shows that *base+ours_train* improves BFCL v3 from 72.09 to 83.65. MMLU-Redux, IFEval, and Multi-IF change from 87.91 to 87.18, 85.95 to 83.55, and 84.69 to 84.28, respectively. These

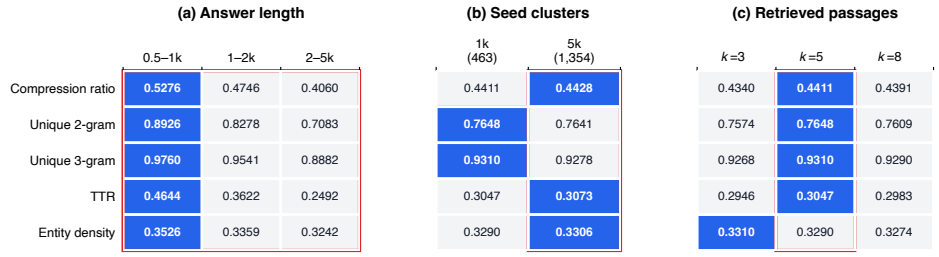

Figure 4: Corpus diagnostics for (a) answer length, (b) seed cluster count, and (c) retrieved-passage count. Cells report raw values, and blue cells mark the best setting for each metric within a panel. Red outlines mark the retained configurations: panel (a) mixes all three answer-length ranges equally, while panels (b) and (c) select 5k seed clusters and $k_r = 5$, respectively. Panels (b) and (c) use 2–5k-character answers, and panel (c) uses the 1k-cluster setting.

supplementary results document the measured capability profile beyond the target educational QA setting.

Model	MMLU-Redux	IFEval	Multi-IF	BFCL v3
base	87.91	85.95	84.69	72.09
base+ours_train	87.18	83.55	84.28	83.65

Table 6: General capability comparison between the base model and our trained model. MMLU-Redux evaluates General Tasks, IFEval (structured prompts) evaluates Alignment Tasks, Multi-IF evaluates Multilingual Tasks, and BFCL v3 reflects Agent & Coding ability.

C Pre-training Diagnostic Definitions

Table 7 specifies the corpus-level diagnostics used to characterize generated answers and to select the reported configuration. These metrics describe surface statistics and do not establish factuality.

D Hyperparameter Experiments and Configuration Selection

These experiments use the pre-training diagnostics in Table 7 to guide configuration selection before downstream SFT. We study answer length, seed cluster count, and retrieved-passage count through conditional one-factor analyses rather than joint optimization.

Answer Length. We compare three final-answer ranges: 500–1,000, 1,000–2,000, and 2,000–5,000 Chinese characters. The requested range is provided to the final generation stage for each configuration. Figure 4(a) shows that the 500–1,000 character setting leads all five diagnostics, confirming their sensitivity to answer length. We therefore control length in method comparisons where possible and mix the three ranges equally in the final SFT corpus.

Seed Cluster Count. We vary the K-Means cluster count among 1k, 3k, 5k, and 10k. After LLM-based extensibility filtering (score threshold 7.0), the retained seed counts are 463, 1,100, 1,354, and 1,587, respectively. The increase becomes smaller from 5k to 10k clusters, suggesting diminishing returns in the number of qualified seeds for this corpus. Figure 4(b) shows nearly identical diagnostics for 1k and 5k clusters. Because 5k retains substantially more qualified seeds (1,354 vs. 463), we use it as the default.

Table 7: Pre-training diagnostic metrics, literature support, and evaluation targets.

Metric	Literature basis and operationalization	Evaluation target
Compression Ratio ↑	Based on the use of lossless compression to capture statistical redundancy in sequences Ziv & Lempel (1977). We compute the gzip compressed size divided by the original byte size for each generated answer.	Surface redundancy and template-like generation; higher values indicate that answers are less compressible under the same length regime.
Unique 2-gram / 3-gram ↑	Following the Distinct-N diversity metric for generated text Li et al. (2016), we compute the proportion of unique character bigrams and trigrams in each generated answer.	Local phrase diversity and avoidance of repetitive wording.
Type-token Ratio (TTR) ↑	TTR is a classical lexical richness measure Templin (1957). We compute the ratio of unique token types to total tokens after Chinese word segmentation with jieba Sun (2012).	Vocabulary coverage and lexical diversity.
Entity Density ↑	Motivated by knowledge-intensive evaluation, where named entities and domain terms are central carriers of factual content Petroni et al. (2021). We extract named entities and technical noun compounds from each answer and report the resulting density score.	Coverage of domain concepts and knowledge-bearing terms.

Retrieved-Passage Count. We compare $k_r \in \{3, 5, 8\}$ retrieved passages per node query. This sweep uses the 1k-cluster condition shown in Figure 4, so it measures retrieval breadth without varying the cluster setting. Figure 4(c) shows that $k_r = 5$ gives the strongest overall profile, leading four of five diagnostics. We therefore use $k_r = 5$ for node grounding.

Selected Configuration. Based on these conditional diagnostic analyses, RiNoGen uses 5k K-Means clusters for seed candidate generation and top-$k_r = 5$ retrieval for node grounding. The selection favors the larger number of qualified seeds at 5k clusters and the strongest diagnostic profile at $k_r = 5$, rather than a jointly optimized downstream-accuracy result.