A APPENDIX

A.1 Dataset

We conduct experiments on 2-hop citation networks for each literature review paper rather than randomly collecting papers to construct a large citation network as a database. This is because using a large, random network complicates performance evaluation, making it difficult to assess both retrieval and clustering accuracy. Additionally, if related papers published after the literature review are present in the citation network, the retriever may include these newer papers in generating the review content. This would lead to an unfair comparison when evaluate the generated content against the human-written review.

Citation Network Construction Process. For each literature review, we first extracted its references and constructed a citation tree, with the review paper as the root and its cited papers as the leaves. We then repeated this process for each cited paper, constructing a citation tree for each one. Next, we merged all these trees into a single, large citation network, consolidating any duplicate nodes. To automate this process, we used citation information from arXiv, which provides the LaTeX source code for each paper, including the bib or bbl files. If a paper was available on arXiv, we extracted its .tex file to obtain both the abstract and full text, using these as high-quality text features for the corresponding node. We used the arXiv API to automate this process. For papers not available on arXiv, we used the Google Scholar API to automatically retrieve the abstract, which we used as the text feature for the corresponding node in the citation network. Finally, we removed the node representing the original literature review, leaving 1-hop, 2-hop, and 3-hop citation networks for each review. The mutual citations among references form complex citation networks, averaging 6,658.4 papers and 11,632.9 edges, including isolated papers.

A.2 Implementation

All experiments were conducted on a Linux-based server equipped with 4 NVIDIA A10G GPUs. For 518 review papers, we successfully collected taxonomy trees with hierarchical citation graph clustering labels for 313 of the literature reviews. Of these, 200 reviews were used to train the hierarchical citation graph clustering and taxonomy generation module, while the remaining 118 were used to test the performance of the pre-trained hierarchical citation graph clustering model. 318 reviews were reserved for a comprehensive evaluation of review content generation. The number of articles retrieved in retrieval phase was set to 200. The scaling factor α is set to 1.

Pre-Train Hierarchical Citation Graph Clustering Module. GNN used in this paper is GAT [45] which has 2 layers with 4 heads per layer and a hidden dimension size of 1024. MLP_ϕ has 2 layers and a hidden dimension size of 1024. The edge connection threshold p_τ is searched in [0.1, 0.2, 0.5, 0.8]. The clustering model is trained for a maximum of 500 epochs using an early stop scheme with patience of 10. The learning rate is set to 0.001. The training batch is set to 512 and the test batch is 1024.

Fine-Tuning. The LLM backbone is Llama-2-7b-hf. We adopt Low Rank Adaptation (LoRA) [13] for fine-tuning, and configure the LoRA parameters as follows: the dimension of the low-rank matrices is set to 8; the scaling factor is 16; the dropout rate is 0.05. For optimization, the AdamW optimizer is used. The initial learning rate is set to 1e-5 and the weight decay is 0.05. Each experiment is run for a maximum of 10 epochs, with a batch size of 4 for both training and testing. The MLP_Φ has 2 layers and a hidden dimension size of 1024.

LLMs. When calling the API, we set temperature as 1 and other parameters to default. The content generator is gpt-4o-2024-05-13 and the content judge is and claude-3-haiku-20240307.

A.3 Evaluation Metrics

We engaged two groups of human raters, each consisting of two professional PhD students in computer science, to evaluate the generated taxonomies and literature reviews using two scoring mechanisms. Each group scores 100 generated samples Given the need to compare model-generated outputs with those written by humans, we designed the evaluation criteria to be straightforward and focused. The two key metrics used are **Adequacy** and **Validity**:

Adequacy is a binary metric where evaluators respond with *"Yes"* or *"No"* to the question: *"Compared to the taxonomy written by humans, is this taxonomy suitable for learning this field?"* This assesses whether the taxonomy is practically usable and meets the fundamental requirements for understanding the domain.

Validity, on the other hand, is rated on a scale from 1 to 5, evaluating the degree to which the taxonomy accurately reflects factual information and represents the domain's conceptual structure. The scoring is defined as follows:

- **1** – Completely inaccurate, with significant factual errors or misrepresentations of the domain.
- **2** – Mostly inaccurate, capturing only a few correct facts but failing to represent the domain coherently.
- **3** – Moderately accurate, containing some factual correctness but missing important concepts or relationships.
- **4** – Mostly accurate, representing the domain well with minor factual inaccuracies or omissions.
- **5** – Highly accurate, thoroughly reflecting the domain's factual structure with no noticeable errors.

This combination of metrics allows us to capture both the *practical usability* and the *factual correctness* of the generated taxonomies, ensuring a comprehensive and nuanced evaluation from multiple perspectives.

For LLM evaluation, we assess the generated content from three perspectives: coverage, relevance, and structure, with each scored on a scale from 1 to 100. The specific prompts used for these evaluations are shown in Figure 5, Figure 6 and Figure 7.

A.4 Investigation of Retrieval Models

We experimented with different retrieval models and strategies, testing two representative methods: the sparse retrieval model, BM25 [33], and the dense retrieval model, SentenceBert [31]. In citation networks, neighbor information and the topological structure play a crucial role in retrieval, as papers on the same topic often cite each other. To assess the impact of using neighbor information, we applied two retrieval strategies for both models: one incorporating

Prompt for evaluating the coverage of generated review.

Instruction: You are an expert in literature review evaluation, tasked with comparing a generated literature review to a human-written literature review on the topic of **[TOPIC]**.

Human-Written Literature Review (Gold Standard):
[GROUND TRUTH REVIEW]

Generated Literature Review (To be evaluated):
[GENERATED REVIEW]

Evaluation Requirements: The human-written literature review serves as the gold standard. Your job is to assess how well the generated literature review compares in terms of coverage. Carefully analyze both reviews and provide a score.

Evaluate Coverage (Score out of 100). Assess how comprehensively the generated review covers the content from the human-written review. Consider:

- The percentage of key subtopics addressed from the human-written review.
- The depth of discussion for each subtopic compared to the human-written version.
- Balance between different areas within the topic as presented in the human-written review.

Only return only a numerical score out of 100, where 100 represents perfect alignment with the human-written literature review, without providing any additional information.

Figure 5: Prompt used for evaluating coverage with LLMs.

Prompt for evaluating the relevance of generated review.

Instruction: You are an expert in literature review evaluation, tasked with comparing a generated literature review to a human-written literature review on the topic of **[TOPIC]**.

Human-Written Literature Review (Gold Standard):
[GROUND TRUTH REVIEW]

Generated Literature Review (To be evaluated):
[GENERATED REVIEW]

Evaluation Requirements: The human-written literature review serves as the gold standard. Your job is to assess how well the generated literature review compares in terms of relevance. Carefully analyze both reviews and provide a score.

Evaluate Relevance (Score out of 100). Evaluate how well the generated literature review aligns with the focus and content of the human-written literature review. Consider:

- Alignment with the core aspects of **[TOPIC]** as presented in the human-written literature review.
- Relevance of examples and case studies compared to those in the human-written literature review.
- Appropriateness for the target audience as demonstrated by the human-written literature review.
- Exclusion of tangential or unnecessary information not present in the human-written version.

Only return only a numerical score out of 100, where 100 represents perfect alignment with the human-written literature review, without providing any additional information.

Figure 6: Prompt used for evaluating relevance with LLMs.

neighbor information as described in Section **??** (*Retrieval w/ Neighbor*) and the other excluding neighbor information (*Retrieval w/o Neighbor*). Given a topic (specifically, the title of a review paper), we retrieved papers related to this topic from the citation network and measured the accuracy by calculating how many of the retrieved papers appeared in the references of the corresponding literature review. The number of retrieved papers was not fixed, but matched the reference count for each review paper.

As shown in Table 5, SentenceBert consistently outperforms BM25 across all scales when neighbor information is not used. For example, in the *1-hop merged* case, SentenceBert achieves an accuracy of 0.5234, significantly higher than BM25's 0.3308. However, both methods show relatively low accuracy without neighbor information, and their performance declines as the size of citation

Prompt for evaluating the structure of generated review.

Instruction: You are an expert in literature review evaluation, tasked with comparing a generated literature review to a human-written literature review on the topic of **[TOPIC]**.

Human-Written Literature Review (Gold Standard):
[GROUND TRUTH REVIEW]

Generated Literature Review (To be evaluated):
[GENERATED REVIEW]

Evaluation Requirements: The human-written literature review serves as the gold standard. Your job is to assess how well the generated literature review compares in terms of structure. Carefully analyze both reviews and provide a score.

Evaluate Structure (Score out of 100). Assess how well the generated literature review's organization and flow match that of the human-written literature review. Consider:

- Similarity in logical progression of ideas.
- Presence of a clear hierarchy of sections and subsections comparable to the human-written literature review.
- Appropriate use of headings and subheadings in line with the human-written version.
- Overall coherence within and between sections relative to the human-written literature review.

Only return only a numerical score out of 100, where 100 represents perfect alignment with the human-written literature review, without providing any additional information.

Figure 7: Prompt used for evaluating structure with LLMs.

Table 5: Results of retrieval on the citation network corresponding to 50 review papers. *2-hop* and *3-hop* represent citation networks of review papers at different scales. *1-hop (merged)* refers to the 1-hop citation network of a review paper, merged with all other 1-hop citation networks, different review papers. Similarly, *2-hop (merged)* is constructed by merging the 2-hop citation network of a review with all other 49 review papers.

Model	Accuracy↑			
	1-hop (merged)	2-hop	2-hop (merged)	3-hop
Retrieval w/o Neighbor				
BM25	0.3308	0.1375	0.0947	0.1014
SentenceBert	0.5234	0.1746	0.1521	0.1490
Retrieval w Neighbor				
BM25	0.7445	0.6435	0.5950	0.6179
SentenceBert	0.2602	0.2758	0.2181	0.2144

networks increases, indicating that retrieving relevant papers becomes more challenging as the network expands. In contrast, BM25 significantly outperforms SentenceBert when neighbor information is utilized. For instance, in the *1-hop merged* case, BM25 reaches an accuracy of 0.7445, while SentenceBert's accuracy drops sharply to 0.2602. BM25 maintains much higher accuracy across all scales with neighbor information. BM25, as a sparse retrieval model, relies on exact term matches, which is particularly advantageous in structured environments like citation networks, where specific terms (e.g., paper titles or keywords) are highly relevant. The inclusion of neighbor information allows BM25 to better capture relationships between papers by focusing on direct term matches in titles or citations. When neighbor information is introduced, the context around the target paper becomes more critical. BM25 effectively leverages this by prioritizing exact matches from neighboring papers, while SentenceBert, which focuses on semantic similarity, may lose precision when handling a broader context that includes less directly related papers.

Without graph-aware retrieval, methods like AutoSurvey must retrieve a large number of papers (e.g., 1200 in AutoSurvey) to avoid missing relevant ones. Retrieving fewer papers risks missing important content, while retrieving too many introduces noise

from irrelevant papers. Graph-aware retrieval significantly alleviates this issue. The graph context-aware retrieval strategy we propose achieves more accurate results with fewer retrievals, i.e., 200, reducing irrelevant information and contributing to the superior generation performance of our model. Moreover, even when applied to large citation networks (such as *2-hop merged* each containing over 200,000 papers), our method maintains stable retrieval accuracy, demonstrating HiReview's robustness across different citation network sizes. Additionally, we experimented with different retrieval strategies, such as retrieving based on both the title and abstract. We found that using only the title yielded the best results, while incorporating additional information like the abstract reduced retrieval performance.

A.5 The Choice of GNN

In addition to GAT [45], we also explored other GNNs as graph encoders, i.e., GCN [16] and Graph Transformer [38]. The comparison results of these models on clustering are shown in Table 6.

Table 6: Performance of different GNN on hierarchical citation graph clustering.

Model	Level 1	Level 2	Average
GAT	0.7127	0.6395	0.6761
GCN	0.6730	0.5963	0.6347
Graph Transformer	0.6811	0.6024	0.6418

GAT achieves the highest performance across both levels, with an average score of 0.6761. It outperforms the other models at both Level 1 (0.7127) and Level 2 (0.6395), making it the most effective GNN for this task. This superior performance can likely be attributed to GAT's attention mechanism, which enables the model to assign varying importance weights to neighboring papers, allowing it to better capture the hierarchical structure of the graph. As a result, we selected GAT as the GNN backbone for HiReview.

B LITERATURE REVIEW GENERATION EXAMPLE

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009

Example Generation for Continual Text Classification.

Continual Text Classification is a crucial area of research in lifelong learning for large language models, addressing the challenge of adapting to new categories and evolving data distributions over time. This field is particularly relevant in real-world applications where new topics or intents may emerge, such as during global events that introduce novel categories. The techniques employed in continual text classification can be broadly categorized into several approaches, including distillation-based methods, replay-based methods, regularization-based methods, and architecture-based methods.

Distillation-based methods focus on transferring knowledge from previous models to new ones. Some approaches use contrastive ensemble distillation to enhance the discriminability between text categories [1], while others employ multi-strategy rebalancing to tackle class imbalance issues [2]. Replay-based methods involve revisiting previously learned information to maintain performance on older tasks. These methods often integrate contrastive learning techniques to enhance the quality of replay samples [3] or use structured learning methods to improve the stability of the learning process [4].

Regularization-based techniques aim to constrain the model's parameter updates to prevent forgetting. Some approaches focus on selectively updating a small subset of parameters to reduce computational burden [5], while others implement adaptive regularization processes that automatically adjust to balance retaining old knowledge with acquiring new information [6]. Architecture-based methods modify the model structure to accommodate new tasks, with some proposing hierarchical overlay projections to optimize knowledge transfer across tasks [7] and others introducing mechanisms for parameter isolation to minimize interference between new and old tasks [8].

In addition to these primary approaches, researchers have explored data augmentation and generation techniques, using large language models to generate diverse and relevant samples for memory augmentation [9]. Meta-learning frameworks, although less common in text classification, have also been employed to enable quick adaptation to new tasks while mitigating catastrophic forgetting [10].

The application of these techniques has been explored in specific domains, such as medical intent detection [11] and social media content analysis [12]. As the field progresses, there is an increasing focus on developing methods that can handle few-shot scenarios [13] and multilingual settings [14]. These advancements aim to make continual text classification more practical and adaptable in real-world applications where data may be scarce or linguistically diverse, ultimately contributing to the development of more robust and versatile large language models capable of continuous learning and adaptation.

[1] CLASSIC: Continual and Contrastive Learning of Aspect Sentiment Classification Tasks
[2] Lifelong intent detection via multi-strategy rebalancing
[3] InfoCL: Alleviating Catastrophic Forgetting in Continual Text Classification from An Information Theoretic Perspective
[4] Class Lifelong Learning for Intent Detection via Structure Consolidation Networks
[5] Parameter-efficient Continual Learning Framework in Industrial Real-time Text Classification System
[6] Hyperparameter-free Continuous Learning for Domain Classification in Natural Language Understanding
[7] HOP to the Next Tasks and Domains for Continual Learning in NLP
[8] Prompts Can Play Lottery Tickets Well: Achieving Lifelong Information Extraction via Lottery Prompt Tuning
[9] Making Pre-trained Language Models Better Continual Few-Shot Relation Extractors
[10] Meta-Learning Improves Lifelong Relation Extraction
[11] Incremental intent detection for medical domain with contrast replay networks
[12] Lifelong Learning of Hate Speech Classification on Social Media
[13] Continual few-shot intent detection
[14] Learning to solve NLP tasks in an incremental number of languages

Figure 8: Example of a generated section on the topic of Continual Text Classification.