Structural Entropy Guided Agent for Detecting and Repairing Knowledge Deficiencies in LLMs

Yifan Wei[1,2], Xiaoyan Yu[3], Tengfei Pan[2], Angsheng Li[1]†, Li Du[2]†
[1]State Key Laboratory of CCSE,School of Computer Science and Engineering,Beihang University
[2]Beijing Academy of Artificial Intelligence, [3]Beijing Institute of Technology
weiyifan@buaa.edu.cn, angsheng@buaa.edu.cn, duli@baai.ac.cn

Abstract

Large language models (LLMs) have achieved unprecedented performance by leveraging vast pretraining corpora, yet their performance remains suboptimal in knowledge-intensive domains such as medicine and scientific research, where high factual precision is required. While synthetic data provides a promising avenue for augmenting domain knowledge, existing methods frequently generate redundant samples that do not align with the model's true knowledge gaps. To overcome this limitation, we propose a novel **S**tructural **En**tropy-guided Knowledge Navig**ator** (SENATOR) framework that addresses the intrinsic knowledge deficiencies of LLMs. Our approach employs the Structure Entropy (SE) metric to quantify uncertainty along knowledge graph paths and leverages Monte Carlo Tree Search (MCTS) to selectively explore regions where the model lacks domain-specific knowledge. Guided by these insights, the framework generates targeted synthetic data for supervised fine-tuning, enabling continuous self-improvement. Experimental results on LLaMA-3 and Qwen2 across multiple domain-specific benchmarks show that SENATOR effectively detects and repairs knowledge deficiencies, achieving notable performance improvements. The code and data for our methods and experiments are available at https://github.com/weiyifan1023/senator.

1 Introduction

With the pretraining process on massive-scale corpora, Large Language Models (LLMs) capture abundant knowledge and demonstrate impressive performance on various downstream tasks (Chen et al., 2015; Liu et al., 2021). However, their performance may still be unsatisfactory in certain knowledge-intensive domains such as medicine and scientific research. This is primarily due to the difficulty in acquiring and scaling up high-quality domain-specific corpora (Lu et al., 2024; Wang et al., 2024), which hinders the ability of the models to handle tasks that require high factual precision.

The development of data synthesis technology (Wang et al., 2023; Zhao et al., 2024) offers an alternative way to address these limitations in remedying the knowledge deficiency of LLMs. While promising, the efficiency of data synthesis remains a significant challenge. This is because current data synthesis methods may not consider the model's knowledge boundaries (Jiang et al., 2021; Mallen et al., 2023; Yue et al., 2025), resulting in substantial efforts spent in generating data that the model may already be familiar with. In fact, even with advanced prompt engineering (Wei et al., 2022; Liu et al., 2025), generated outputs tend to skew toward high-frequency distributions seen in pretraining data, leading to severe redundancy. Therefore, efficient data synthesis should be tightly coupled with mechanisms for effectively detecting knowledge deficiencies (Xiong et al., 2024; Song et al., 2025) within LLMs, so that the synthesized data can repair the knowledge deficiencies.

†Corresponding Authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

However, the knowledge boundaries of large models can be quite complex. Although these models are trained on massive amounts of data, their knowledge is implicitly encoded in model parameters (Geva et al., 2021; Wei et al., 2025) rather than being explicitly stored, leading to unclear distinctions between known and unknown information. In specialized domains, this challenge is compounded by the generation of unreliable or contradictory content (Yang et al., 2024c), which produces flawed synthetic samples that hinder the effective expansion of high-quality, domain-specific corpora.

To overcome the aforementioned challenges, we propose SENATOR, a **S**tructural **En**tropy-guided Knowledge Navig**ator** framework, which achieves knowledge deficiency remediation through a closed loop of structured knowledge probing and targeted synthetic data generation. The framework comprises two key components: 1) Knowledge Deficiency Detection: Human-annotated knowledge graph (KG) systematically describes the underlying complexities and intricacies of the domain. However, the combinatorial explosion of possible paths makes enumeration computationally infeasible. To efficiently detect the knowledge paths, we drive the LLM as an agent to explore upon the KG in a Monte Carlo Tree Search (MCTS) manner (Metropolis and Ulam, 1949), with the structure entropy as reward. The Structure Entropy (SE) (Li and Pan, 2016; Li, 2024) metric quantifies the structural information contained within a graph by capturing its topological organization and the interactions among nodes. This provides insight into the model's uncertainty along knowledge paths in the KG. By employing MCTS within the knowledge space, our framework uses SE values as intrinsic rewards to decide whether to expand specific entity nodes, effectively prioritizing the exploration of paths with high uncertainty and detecting critical knowledge deficiencies. 2) Knowledge Synthesis and Repair: Leveraging the critical knowledge paths identified via MCTS, our framework generates synthetic data by employing prompt templates to structure the content. The KG serves as a trusted source to ensure both the data inputs and the synthesized outputs are credible and contextually relevant. This synthetic data is then used to fine-tune the model through supervised learning, enabling continuous self-improvement and effective remediation of knowledge deficiencies.

Our experiments demonstrate that the SENATOR framework effectively detects knowledge deficiencies in large language models and efficiently repairs them, leading to significant performance improvements across multiple domain-specific benchmarks. Data distribution analyses confirm that our synthetic data incorporates knowledge deficiencies from the pretraining corpus. Moreover, supervised fine-tuning (SFT) of LLMs like Llama-3 (Grattafiori et al., 2024) and Qwen2 (Yang et al., 2024b) using this data led to significant performance improvements, demonstrating that targeted injection of missing knowledge can substantially enhance overall model performance.

2 Related Work

Knowledge Deficiency Detection of LLMs Though LLMs possess extensive knowledge, they often struggle to accurately delineate what they know from what they do not (Yin et al., 2023; Ren et al., 2023). Several approaches (Jiang et al., 2020; Mallen et al., 2023; Wei et al., 2024)construct knowledge probability distributions based on existing annotated data, using metrics such as answer correctness or confidence scores to assess a model's knowledge proficiency. One line of work (Wei et al., 2022; Li et al., 2023a; Tian et al., 2024a) directly toward enhancing a model's ability to fully leverage its existing knowledge, thereby reducing the proportion of "Unknown Knows". Another line of work pay attention to enabling models to explicitly acknowledge their knowledge gaps, thus minimizing the occurrence of "Unknown Unknowns". Approaches such as R-tuning (Zhang et al., 2023) utilize labeled data with supervised fine-tuning to judge response correctness, while reinforcement learning based strategies have also been explored (Yang et al., 2023b; Kang et al., 2024). In contrast, our approach for deficiency detection is designed not to rely on pre-existing labeled data, but instead to actively explore the KG to detect intrinsic model uncertainty.

Model Self-Improvement Self-improvement methods of LLM focus on leveraging internal knowledge and feedback to iteratively enhance the performance of LLMs (Zelikman et al., 2022, 2024). A pivotal challenge is generating a reliable critique signal to discern high-quality responses from suboptimal ones. Previous methods (Bai et al., 2022; Wang et al., 2023) involve prompting the LLM to generate diverse task-specific queries and corresponding outputs, followed by the application of manually crafted heuristic rules, such as filtering based on query length to remove redundant or low-quality data pairs. Given the complexity of devising effective heuristics, subsequent research (Sun et al., 2023; Li et al., 2023b; Guo et al., 2024) proposes a few general principles or judging

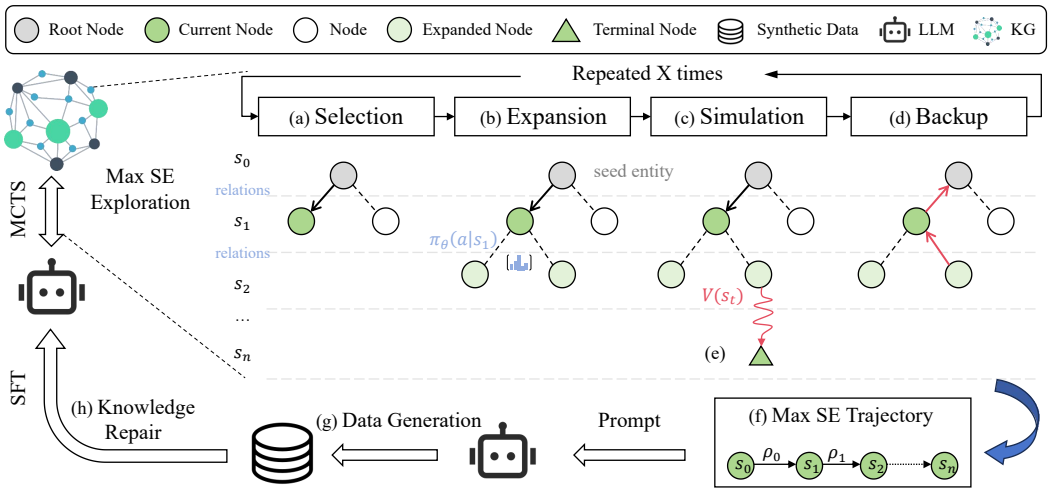


Figure 1: The SENATOR framework operates as follows: An entity state in the knowledge graph is (a) selected, (b) expanded, and (c) simulated using the LLM agent until a terminal node is reached. Specifically, we employ a random policy π during the expansion phase. (d) Subsequently, signals from the value function $V(\cdot)$ are backpropagated. This process is iterated multiple times, with the MCTS algorithm searching for (f) better trajectories guided by (e) signals from structural entropy to (g) generate data addressing knowledge deficiencies, (h) and repair model knowledge.

criteria and ask the LLM itself to assess the quality its responses according to these guidelines. However, this approach demands that LLMs possess a robust capability to apply these principles to each specific instance and render accurate judgments. Recently, reinforcement learning-based model show impressive reasoning ability by learning the experiences obtained from explorations in the solution space (Tian et al., 2024b; Goldie et al., 2025). While the probability of obtained plausible solution space of knowledge intensive tasks would be rather limited as the LLM may not possess the necessary knowledge, which would severely restrict the efficiency of exploration and data generation. In this paper, we choose to guide the exploration process in knowledge space using KGs, in a MCTS manner, so as to enable targeted synthetic data generation for high efficiency LLM self-improvement.

3 Methodology

Given a knowledge graph, the number of possible knowledge paths $\mathcal{P}$ (i.e., Figure 1f) increases in a combinatorial speed along with the size of KG, making enumerating all possible paths and detecting the uncertainty of LLM on these paths computationally infeasible. To tackle this challenge, as shown in Figure 1, SENATOR employs MCTS to navigate the LLM-based agent to search on the KG for seeking out the most informative paths. To steer the agent to search toward regions with high uncertainty, we introduce a structural entropy based reward function. Based on the identified high-uncertainty paths, data are synthesized to remediate the identified knowledge deficiencies.

3.1 Structural Entropy Guided Knowledge Deficiency Detection

The structural entropy based reward function combines the uncertainty of LLM on individual KG triplets with the topological structure information of the KG, guiding the LLM-based agent to perform MCTS over the KG and discover knowledge paths with critical deficiencies.

Self-Information for Measuring Triplet-Level Uncertainty Self-Information (Shannon, 1948) quantifies the amount of information conveyed by a "fact" given its probability distribution. In KGs, a "fact" is represented as a triplet $\tau =<$ subject u, relation ρ, object $v >$. To measure the LLM's uncertainty of such "facts", we transform τ into a cloze statement form. The cloze context is formed by combining the subject u and the relation ρ, creating a prompt to predict the missing object v. The

self-information of a fact τ is defined as:

$$I(u, \rho, v) = -\log_2 P(v \mid u, \rho), \tag{1}$$

where $P(v \mid u, \rho)$ is the probability of the output v conditioned on the cloze context. Since the relation ρ in KGs is directional, the self-information calculated in this manner serves as a measure of the factual knowledge confidence for the entire triplet.

Structural Entropy of Modeling Knowledge Path-Level Uncertainty To integrate the uncertainty of all triplets along a knowledge path while considering their structural importance, we adopt structural entropy (SE) as a more comprehensive measure of an LLM's knowledge confidence, as shown in Figure 1e. Structural importance reflects the topological significance of a triplet τ within the knowledge graph. Triplets involving highly connected entities are considered more central, as these entities participate in more relational paths and exert broader influence across the graph. Unlike self-information or Shannon entropy, structural entropy accounts for the knowledge graph's topological structure and the interdependencies among its elements. This is crucial because each triplet is not an isolated piece of information but part of a structured network. The relationships among entities contribute to the overall representation of knowledge. Given a knowledge graph $G = (V, E)$, each edge $\rho \in E$ is assigned a weight derived from the self-confidence in Equation 1. The weighted degree of an entity node $u \in V$ is defined as:

$$d_u = \sum_{v \in \mathcal{N}(u)} I(u, \rho, v), \tag{2}$$

where $\mathcal{N}(u)$ denotes the set of neighbors of entity u and d_u represents the overall uncertainty contained within the node. To quantify the average information content of the graph G, we define the one-dimensional structural entropy of the weighted, connected graph G as:

$$\mathcal{H}^1(G) = -\sum_{u \in V} \frac{d_u}{\mathrm{vol}(G)} \log_2 \frac{d_u}{\mathrm{vol}(G)}, \tag{3}$$

where $\mathrm{vol}(G)$ represents the total weighted degree of G. A higher $\mathcal{H}^1(G)$ indicates a more complex and less confidently represented region within the knowledge graph. By formulating SE as the exploration reward in MCTS, we enable the search algorithm to prioritize paths traversing maximally uncertain knowledge structures, thereby efficiently exposing the model's systemic weaknesses.

3.2 MCTS for Knowledge Deficiency Detection

Given the SE-based reward function, we employ MCTS to explore the KG and identify potential knowledge deficiency paths in the model. We define the initial state s_0 as the starting node for traversing the KG, where a set of seed entities from (Soman et al., 2024) is selected. KG triplets are incrementally incorporated into the knowledge paths until the maximum search depth T is reached. This process enhances the LLM's awareness of its knowledge deficiencies by maximizing the expected reward, which emphasizes the uncertainty associated with these deficiencies.

Node Selection. The objective of this stage is to identify and prioritize KG entities that are likely to expose the LLM's knowledge deficiencies, as shown in Figure 1a. Formally, at state s_t, the LLM agent reaches entity node u_t of the KG, and the MCTS process choose from $\mathcal{A} = \{a_1, a_2, \ldots, a_m\}$, representing the relation edges ρ_{t+1} that connect the current entity u_t to its neighbors $\mathcal{N}(u_t)$. It is guided by two key variables: $Q(s_t, a)$, the cumulative value of taking action a in state s_t, and $N(s_t)$, the visitation frequency of state s_t. Heuristically, $Q(s_t, a)$ guides exploitation by favoring actions with historically high rewards, while $N(s_t)$ encourages exploration of under-visited states. We integrate these complementary objectives using the PUCT algorithm (Rosin, 2011), which selects the next state as:

$$s^*_{t+1} = \arg\max_{s_t} \left[Q(s_t, a) + c_{\text{puct}} \cdot P\left(a \mid s_t\right) \frac{\sqrt{N\left(s_t\right)}}{1 + N\left(s_t, a\right)} \right], \tag{4}$$

where $P(a|s_t)$ denotes the prior probability of selecting action a given state s_t. In this way, an additional triplet τ is incorporated into the knowledge path $\mathcal{P}$.

Path Expansion. Expansion occurs when a leaf node is reached during the selection phase, enabling the integration of new states and the assessment of immediate rewards. Upon reaching a leaf node, it is expanded by selecting all possible relation action from leaf node, where each action a represents a transition from the current entity state s_t to a new entity state s_{t+1} in $\mathcal{N}(s_t)$, as shown in Figure 1b. These unexplored entities $\mathcal{N}(s_t)$ are then added as leaf nodes to the search tree. The immediate reward function $r(s_t, a)$ quantifies the advantage of each action $a \in \mathcal{A}$ available at state s_t.

$$\begin{aligned} r_{t+1} = r(s_t, a) = I(s_t, a, s_{t+1}) = -\log_2 \frac{d_{s_{t+1}}}{\mathrm{vol}(G)}, \\ V(s_t) = r_{t+1} + \gamma V(s_{t+1}) = \sum_{k=0}^{T-k-1} \gamma^k \, r_{t+k+1}, \end{aligned} \tag{5}$$

where γ is the discount factor for future state values $V(\cdot)$ and T is the depth of the MCTS search space. To accommodate scenarios with limited decision steps and stable reward distributions, we eliminate the discount factor and instead compute the average of future immediate reward values, as formalized in Equation 6.

Reward Estimation. A simulation shown in Figure 1c is run from the new expanded node s_t by making random relation actions until a terminal state is reached. The newly expanded nodes are evaluated using an evaluation function integrating future rewards, state relevance, and actual outcomes. In this paper, we propose a novel intrinsic reward mechanism to address the limitation of Shannon entropy in handling structured data. To overcome this challenge, we define one-dimensional structural entropy as an intrinsic reward for effective exploration:

$$\begin{aligned} V(s_t) = H(\mathcal{P}) &= \mathbb{E}\left[\sum_{k=0}^{T-k-1} r_{t+k+1} \middle| \, s_t \right] \\ &\approx \mathcal{H}^1(\mathcal{G}) = -\sum_{s_t \in \mathcal{P}} \frac{d_{s_t}}{\mathrm{vol}(\mathcal{G})} \log_2 \frac{d_{s_t}}{\mathrm{vol}(\mathcal{G})}, \end{aligned} \tag{6}$$

where $\mathcal{P} = \{s_t, s_{t+1}, \cdots, s_T\}$ denote the selection trajectory of t-th iteration, which ends at the terminal state s_T after one complete simulation. For simplicity, the notation omits the relationships $\mathcal{A}$ between states. Specifically, $\mathcal{G}$ is a subgraph of the knowledge graph G, representing a given search space, and we utilize the structural entropy on this subgraph to approximate the state value.

Backpropagation. We update the statistics of each state in the tree that was traversed during the selection stage. Specifically, the back propagation process updates the value estimates and visit counts of all ancestor nodes along the trajectory $\mathcal{P}$ as shown in Figure 1d, ensuring leaf node evaluation informs higher-level decision-making. The updated rules are as follows:

$$\begin{aligned} N(s_t) &\leftarrow N(s_t) + 1, \\ Q(s_t, a) &\leftarrow \frac{1}{N(s_t, a)} \sum_{i=1}^{N(s_t)} \mathbb{I}_i(s_t, a) V_i(s_t), \end{aligned} \tag{7}$$

where $N(s_t, a)$ is the number of times relation action a has been selected from state s_t, $N(s_t)$ is the number of times a simulation has been run from state s_t, and $\mathbb{I}_i(s_t, a)$ is 1 if relation action a was selected from state s_t on the i-th simulation run from state s_t, or 0 otherwise.

3.3 Deficiency Knowledge Synthesis and Repair

As shown in Figure 1f to 1h, our framework leverages the trajectories with the highest SE values obtained via MCTS to guide synthetic data generation. Specifically, we prompt the LLM agent to synthesize a set of QA pairs based on the identified knowledge path on which the LLM shows high uncertainty, so that the knowledge deficiency of the LLM can be remedied by training on these QA pairs. Formally, as shown in Figures 5 and 6, given a trajectory $\mathcal{P} = \{s_1, s_2, \ldots, s_T\}$, the prompt instructs the LLM to generate a question that focuses on $\mathcal{P}$ and an answer that logically explains on the relationship ρ_{t+1} between s_t and its neighboring entities $\mathcal{N}(s_t)$ in $\mathcal{P}$. So that the synthesized QA pair can adhere to the underlying knowledge about the knowledge path and remedy

the knowledge deficiency of the LLM. Furthermore, to maintain high data quality, we implement a multi-tiered evaluation mechanism that includes both heuristic rules and LLM-based judgments. Our quality standards encompass: *Format Consistency:* The generated QA pairs must strictly adhere to the predefined prompt template, ensuring that the structure, punctuation, and length conform to our specifications. This guarantees that the synthesized data maintains a uniform format that facilitates downstream processing. *Logical Coherence:* The QA pairs must exhibit clear and rational reasoning. The answer should provide a logically consistent explanation that reflects the relationships and context derived from the knowledge trajectory, ensuring that the data effectively captures and addresses the identified knowledge deficiencies. *Hallucination Avoidance:* The generated content must be grounded in the input trajectory. Specifically, all entities and facts mentioned in the QA pair must originate exclusively from the given trajectory, preventing the introduction of extraneous or unsupported information that could undermine the model's reliability. Data samples that do not meet these criteria are filtered out through our evaluation mechanism A.1, thereby ensuring that only high-quality synthetic data is used to remediate the LLM's knowledge gaps.

The training process can be divided into two stages: First, a knowledge injection stage, that aims to enrich the LLMs with deficiency medical knowledge D_K. Second, a medical instruction tuning stage, that tailors the model to align with the medical QA domain. (see Appendix A.3 for details).

4 Experiments

We conduct experiments on the knowledge-intensive *medical domain* to investigate the following research questions (RQs): **RQ1**: Can the proposed SENATOR framework effectively repair the knowledge deficiencies of existing LLMs? **RQ2**: How do different components of our proposed framework impact the performance of LLMs? **RQ3**: Does the synthetic data successfully incorporate knowledge that lies beyond the distribution of the pretraining corpus? **RQ4**: What is the scaling regularity of synthetic data on model performance?

4.1 Experimental Settings

Language Models We evaluate our methodology on two categories of LLMs: 1) General LLMs: We employ Llama-3-8B and Qwen2-7B as base models to examine the effectiveness of our approach and include Baichuan2 and Llama-2 for comparison. 2) Medical LLMs: Med-Alpaca (Han et al., 2023): Fine-tuned on LLaMA-13B with medical instruction data from Alpaca (Han et al., 2023), specifically designed for medical dialogues and question-answering tasks. PMC-LLaMA (Wu et al., 2024): Enhanced with biomedical knowledge from 4.8 million academic papers and 30,000 medical books, followed by medical-specific instruction tuning on LLaMA-13B. HuatuoGPT-II (Chen et al., 2023a): Built on Baichuan (Yang et al., 2023a), fine-tuned with distilled ChatGPT data and real-world medical data from doctors.

Datasets Our instruction tuning data D_I, which contains 514k samples, is derived from Wu et al. (2024) to align with the medical domain. It's widely used in the medical field for its large scale and comprehensive coverage of medical knowledge. We evaluate our approach on five standard medical benchmarks: 1) **MedQA** (Jin et al., 2021): Multiple-choice questions from the USMLE assessing medical understanding and reasoning. 2) **MedMCQA** (Pal et al., 2022): Over 194K questions from AIIMS exams covering 2,400 topics across 21 subjects. 3) **PubMedQA** (Jin et al., 2019): A biomedical QA dataset from PubMed abstracts with 1K expert-annotated and 211K generated QA instances, designed to test comprehension and reasoning in biomedical research. 4) **GPQA** (Rein et al., 2023): A high-difficulty multiple-choice dataset validated by experts in biology, physics, and chemistry, focusing on interdisciplinary knowledge and reasoning. 5) **MMLU** (Hendrycks et al., 2020): A comprehensive benchmark covering 57 tasks for evaluating large language models.

Knowledge Graph We conduct experiments based on the SPOKE knowledge graph (Morris et al., 2023) due to its comprehensiveness on biological and medical knowledge, which contains over 42 million nodes of 28 different types and 160 million edges of 91 types, constructed by integrating information from 41 different biomedical databases. In this paper, the initial seed entities for MCTS are common disease entities in SPOKE, sourced from Soman et al. (2024).

Table 1: Main Results on Medical Benchmarks in the Zero-shot Setting. Δ represents the relative change in performance when using our synthetic data generated by SENATOR compared to the corresponding backbone model. "w/" denote "with" and IT represents instruciton tuning data.

Model	MedQA	MedMCQA	PubMedQA	GPQA		Avg.
				Genetics	Molecular Biology	
Human (pass)	50.0	–	60.0	43.2		–
Human (expert)	87.0	90.0	78.0	66.7		80.43
Medical LLMs						
Chat-Doctor (7B)	33.93	31.10	54.3	–	–	–
Med-Alpaca (13B)	30.85	31.13	53.2	10.0	15.43	28.12
HuatuoGPT-II (7B)	41.13	41.87	54.2	22.5	21.60	36.26
HuatuoGPT-II (13B)	45.72	38.75	51.6	20.0	27.78	36.77
PMC-LLaMA (13B)	50.67	50.18	59.8	15.0	27.16	40.56
General LLMs						
Baichuan2-7B	34.56	35.12	60.2	20.0	20.99	34.17
Baichuan2-13B	43.60	39.25	50.7	27.5	30.86	38.38
Llama-2-7B	30.95	28.85	60.8	25.0	17.28	32.58
Llama-2-13B	31.26	29.00	62.2	35.0	20.99	35.69
Llama-3-8B	55.54	52.21	54.8	20.0	29.01	42.31
w/ instruction tuning	54.36	50.08	56.6	25.0	25.93	42.39
w/ synthetic data + IT	58.29	53.60	64.8	27.5	32.72	47.38
Δ promotion	+4.95%	+2.66%	+18.25%	+37.50%	+12.79%	+11.98%
Qwen2-7B	54.67	53.41	64.6	32.5	36.42	48.32
w/ instruction tuning	59.07	59.77	61.2	22.5	35.80	47.67
w/ synthetic data + IT	59.70	60.70	63.2	40.0	40.12	52.74
Δ promotion	+9.20%	+13.65%	-2.17%	+26.08%	+10.16%	+9.15%

4.2 Main Results (RQ1)

Table 1 presents the performance of our approach and baseline models across four medical benchmarks. From this, we observe that (1) Through continuous pretraining on medical corpora, previous medical domain LLMs such as PMC-LLaMA could achieve ordinary-human-level performance on certain benchmarks. For example, **PMC-LLaMA employs approximately 514k samples, 79 billion tokens of medical data** to achieve performances close to such as MedQA and PubMedQA. However, its performance on genetics-related subset of GPQA still shows a substantial gap with human-level, indicating significant knowledge deficiency. (2) In contrast, our proposed SENATOR framework demonstrates its effectiveness in finding knowledge deficiencies to efficiently adapt LLMs to the medical domain. When applied to Llama-3-8B and Qwen2-7B, the SENATOR framework uses a much smaller amount of synthetic data **(26k samples, 0.8 million tokens and 128k samples, 3.6 million tokens, respectively)** to remedy the targeted knowledge areas, and improve the performance on corresponding benchmarks. For instance, the SENATOR optimizes the Qwen2 model attains an accuracy of 40% on the Genetics component of GPQA, demonstrating that supplementing missing domain-specific data can substantially enhance performance. Overall, on the four medical domain-related benchmarks, on average, the SENATOR framework improves the performance of Llama-3-8B and Qwen2-7B for 11.98% and 9.15%, respectively. This shows the effectiveness and generality of our approach in comprehensively detecting and remedying the domain-related knowledge for different LLMs. In the following paragraphs (RQ2 and RQ3), we demonstrate that the improvement stems from SENATOR's ability to effectively detect the knowledge deficiencies by synthesizing data beyond the original pretraining corpus, expanding its coverage, and optimizing its distribution.

4.3 Ablation Study (RQ2)

To validate the efficacy of SENATOR, we conduct ablation studies comparing three configurations: (1) base models, (2) models fine-tuned solely with general domain instruction data D_I, and (3) models

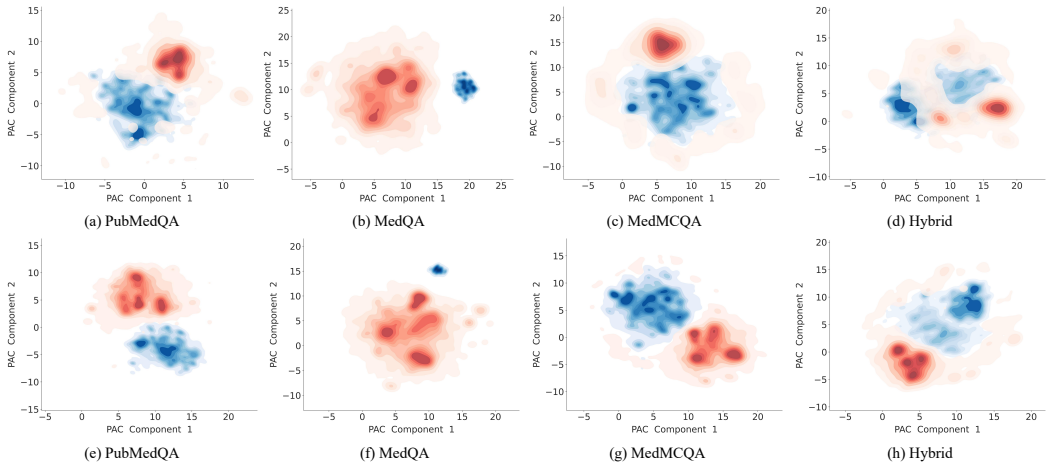


Figure 2: Distribution of Pretraining Corpus vs. Synthetic Data. In (a)-(d), blue regions represent the medical pretraining corpus (PubMedQA, MedQA, MedMCQA, and their hybrid), red regions show synthetic data generated by Llama-3. In (e)-(h), red regions indicate synthetic data produced by Qwen2. Darker areas reflect higher concentrations of data points, lighter areas vice versa.

trained with both instructions and synthesized data. As shown in Table 1, SFT on general domain instructions alone yields marginal improvements or even performance degradation (Llama-3-8B: 42.31 $\rightarrow$ 42.39; Qwen2-7B: 48.32 $\rightarrow$ 47.67). This suggests that the general domain instructions struggle to alleviate the intrinsic knowledge gaps in general-domain LLMs for the specialized medical domain, and constructing more general domain instructions would inevitably be inefficient. In contrast, incorporating synthetic data leads to a significant improvement. For Llama-3-8B, additional synthesized data make average performance improvements of 5.07, with particularly significant gains in underrepresented domains: +7.5 points in GPQA Genetics and +3.71 points in Molecular Biology. Similarly, Qwen2-7B attains 40.0% accuracy in GPQA Genetics (7.5-point increase) and 40.12% in Molecular Biology (3.7-point gain). These results indicate that performance improvement is brought by synthesizing data from detecting the deficiency of LLMs instead of simply enlarging the size of existing instruction data, and **a deficiency-oriented synthetic data generation strategy** would be a more efficient method for expanding knowledge of LLMs, suggesting a way towards "new fuel" (PwC Australia, 2023) for enriching the existing corpus and empowering future LLMs.

4.4 Analysis for Distribution of Synthesized Data (RQ3)

To examine if our approach can generate synthetic data beyond the original pretraining distribution and address the knowledge deficiency of LLMs, we visualize the distribution of both the original pretraining data, which is sourced from the training sets of PubMedQA, MedQA, and MedMCQA, and the synthetic data. This visualization is achieved by first projecting data into a unified semantic space using 2D UMAP (McInnes et al., 2018) and obtaining their distribution using kernel density estimation (KDE) (Rosenblat, 1956; Parzen, 1962). From Figure 2 we can observe: **1) Expanded Coverage by synthetic data:** Figures 2a to 2h reveal that the red area (representing synthetic data) encircles the blue area (pretraining data), indicating that the synthetic data effectively broadens the coverage of the pretraining data. Additionally, Figure 2b and 2f display smaller blue regions, indicating that the distribution of synthesized data is much broader than the pretraining data available for MedQA. **2) Distribution Overlap:** In Figure 2d, the synthetic data

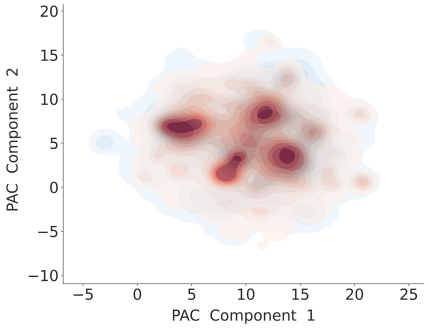
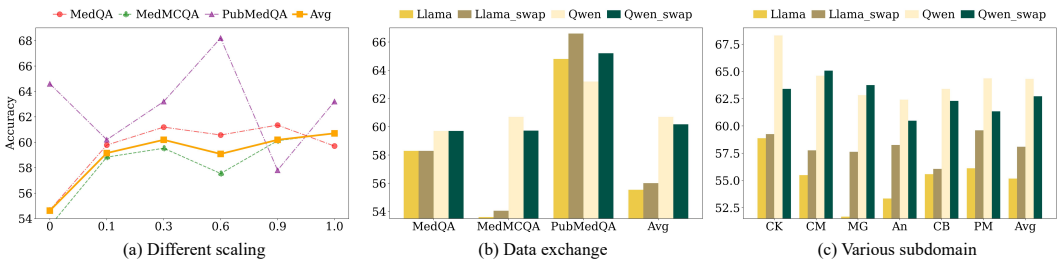


Figure 3: Distribution of Data Generated by Llama-3 (red) and Qwen2 (blue).



(a) Different scaling (b) Data exchange (c) Various subdomain

Figure 4: Performance differences for various data compositions.

shows a high degree of overlap with the overall pretraining data. We hypothesize that this may be due to Llama-3's relatively weaker grasp of pretraining knowledge compared to Qwen2, causing SENATOR to collect information that Llama-3 did not consolidate well during pretraining. **3) Topic-Specific Differences:** Compared to Figure 2a, Figure 2e exhibits an opposite trend. Accordingly, as indicated in Table 1, Qwen2 demonstrates a higher performance on PubMedQA. This is likely because Qwen2 demonstrated a stronger mastery of PubMedQA during pretraining (Yang et al., 2024a), leading SENATOR to explore that topic distribution to a lesser extent during the defect detection phase. **4) Global Trends and Localized Discrepancies:** The analysis of synthetic data distributions generated by Llama-3 and Qwen2 (Figure 3) shows substantial overlap in high-density areas, indicating that both models have a roughly similar pattern (may also share with more LLMs) in knowledge deficiency about the medical domain. This is because of the similarity in the distribution of the pretraining corpus (Lee et al., 2022; Yauney et al., 2023). Such similarity indicates the necessity of systematically reviewing the deficiencies of present LLMs to find common knowledge blind spots in the pretraining corpus, and synthesizing data to complement them. However, there still exist differences in certain locations, suggesting model-specific knowledge deficiencies. This suggests the effectiveness of our approach in targeting model-specific knowledge deficiencies.

4.5 Analysis of Synthetic Data Scaling (RQ4)

To explore how the amount of synthetic data affects model repair, we integrate different proportions of synthetic data into the SFT stage, as depicted in Figure 4a. We observe an upward trend in overall performance, calculated as a weighted average based on dataset sizes, with increasing synthetic data proportions. This indicates that, when the instruction-aligned data D_I is fixed, expanding the synthetic data enhances model performance. As more synthetic data is used, more LLM knowledge deficiencies can be identified and addressed, thereby improving the model's performance. This highlights the potential of our method to effectively boost model performance by targeting and synthesizing data to fill specific knowledge gaps. Due to the limitation in computation resources, in this paper, for the two base LLMs, Llama and Qwen, we synthesize 26k and 128k data entries, respectively. In future work, we will explore integrating diverse knowledge across more domains to further enhance model performance. Additionally, we compare two settings: the default setting (SENATOR), where each model is fine-tuned using data synthesized using its own detected deficiencies, and the swap setting, where a model is trained with data synthesized using deficiencies of another model, for example, synthetic data produced by Llama-3 is used for SFT of Qwen2, and vice versa. As shown in Figure 4b and 4c, SENATOR demonstrates effective deficiency correction even under the swap setting. This could be brought by the similarities between the pretraining corpus of different LLMs, which can lead to similar knowledge deficiencies. This finding not only reinforces the potential of our synthetic data as a valuable supplement to human-written corpora, but also highlights the pressing need for efficient and comprehensive strategies to detect and repair knowledge deficiencies in LLMs.

5 Conclusion

In this paper, we introduce SENATOR, an innovative framework that utilizes structural entropy and knowledge graphs to detect and repair knowledge deficiencies in LLMs. By employing MCTS within the knowledge space, SENATOR effectively identifies areas where the model's understanding is deficient. Leveraging the SENATOR agent, we direct the synthetic data generation process to

specifically target these deficiencies. Our experiments on medical benchmarks reveal significant performance improvements when models like Llama-3 and Qwen2 are fine-tuned with the synthetic dataset. These results highlight that a deficiency-oriented synthetic data generation strategy represents a highly efficient and sustainable method for expanding knowledge, positioning it as the "new fuel" of modern AI.

6 Acknowledgements

We thank the support of the National Science and Technology Major Project (2022ZD0116301), the General Program of the National Natural Science Foundation of China (62576021) and the Youth Fund of the National Natural Science Foundation of China (62406040).

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Junying Chen, Xidong Wang, Ke Ji, Anningzhe Gao, Feng Jiang, Shunian Chen, Hongbo Zhang, Dingjie Song, Wenya Xie, Chuyi Kong, et al. 2023a. Huatuogpt-ii, one-stage training for medical adaption of llms. *arXiv preprint arXiv:2311.09774*.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176.

Zeming Chen, Alejandro Hernández-Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. 2023b. Meditron-70b: Scaling medical pretraining for large language models. *Preprint*, arXiv:2311.16079.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495.

Anna Goldie, Azalia Mirhoseini, Hao Zhou, Irene Cai, and Christopher D Manning. 2025. Synthetic data generation & multi-step rl for reasoning & tool use. *arXiv preprint arXiv:2504.04736*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Hongyi Guo, Yuanshun Yao, Wei Shen, Jiaheng Wei, Xiaoying Zhang, Zhaoran Wang, and Yang Liu. 2024. Human-instruction-free llm self-alignment with limited samples. *arXiv preprint arXiv:2401.06785*.

Tianyu Han, Lisa C Adams, Jens-Michalis Papaioannou, Paul Grundmann, Tom Oberhauser, Alexander Löser, Daniel Truhn, and Keno K Bressem. 2023. Medalpaca–an open-source collection of medical conversational ai models and training data. *arXiv preprint arXiv:2304.08247*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577.

Katie Kang, Eric Wallace, Claire Tomlin, Aviral Kumar, and Sergey Levine. 2024. Unfamiliar finetuning examples control how language models hallucinate. *arXiv preprint arXiv:2403.05612*.

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. 2024. Biomistral: A collection of open-source pretrained large language models for medical domains. *Preprint*, arXiv:2402.10373.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445.

Angsheng Li. 2024. *Science of Artificial Intelligence: Mathematical Principles of Intelligence (In Chinese)*. Science Press, Beijing.

Angsheng Li and Yicheng Pan. 2016. Structural information and dynamical complexity of networks. *IEEE Transactions on Information Theory*, 62(6):3290–3339.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023a. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, volume 36, pages 41451–41530. Curran Associates, Inc.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Luke Zettlemoyer, Omer Levy, Jason Weston, and Mike Lewis. 2023b. Self-alignment with instruction backtranslation. *arXiv preprint arXiv:2308.06259*.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2021. Logiqa: a challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3622–3628.

Kai Liu, Ze Chen, Zhihang Fu, Rongxin Jiang, Fan Zhou, Yaowu Chen, Yue Wu, and Jieping Ye. 2025. Structure-aware domain knowledge injection for large language models. pages 29443–29464.

Zimu Lu, Aojun Zhou, Houxing Ren, Ke Wang, Weikang Shi, Junting Pan, Mingjie Zhan, and Hongsheng Li. 2024. Mathgenie: Generating synthetic data with question back-translation for enhancing mathematical reasoning of llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2732–2747.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861.

Nicholas Metropolis and Stanislaw Ulam. 1949. The monte carlo method. *Journal of the American statistical association*, 44(247):335–341.

John H Morris, Karthik Soman, Rabia E Akbas, Xiaoyuan Zhou, Brett Smith, Elaine C Meng, Conrad C Huang, Gabriel Cerono, Gundolf Schenk, Angela Rizk-Jackson, et al. 2023. The scalable precision medicine open knowledge engine (spoke): a massive knowledge graph of biomedical information. *Bioinformatics*, 39(2):btad080.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.

Emanuel Parzen. 1962. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076.

PwC Australia. 2023. Synthetic data: The new fuel for ai.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, Jing Liu, Hao Tian, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2023. Investigating the factual knowledge boundary of large language models with retrieval augmentation. *arXiv preprint arXiv:2307.11019*.

M Rosenblat. 1956. Remarks on some nonparametric estimates of a density function. *Ann. Math. Stat*, 27:832–837.

Christopher D Rosin. 2011. Multi-armed bandits with episode context. *Annals of Mathematics and Artificial Intelligence*, 61(3):203–230.

Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Karthik Soman, Peter W Rose, John H Morris, Rabia E Akbas, Brett Smith, Braian Peetoom, Catalina Villouta-Reyes, Gabriel Cerono, Yongmei Shi, Angela Rizk-Jackson, et al. 2024. Biomedical knowledge graph-optimized prompt generation for large language models. *Bioinformatics*, 40(9):btae560.

Linxin Song, Xuwei Ding, Jieyu Zhang, Taiwei Shi, Ryotaro Shimizu, Rahul Gupta, Yang Liu, Jian Kang, and Jieyu Zhao. 2025. Discovering knowledge deficiencies of language models on massive knowledge base. *arXiv preprint arXiv:2503.23361*.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. 2023. Principle-driven self-alignment of language models from scratch with minimal human supervision. *arXiv preprint arXiv:2305.03047*.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. 2024a. Fine-tuning language models for factuality. In *The Twelfth International Conference on Learning Representations*.

Ye Tian, Baolin Peng, Linfeng Song, Lifeng Jin, Dian Yu, Lei Han, Haitao Mi, and Dong Yu. 2024b. Toward self-improvement of llms via imagination, searching, and criticizing. *Advances in Neural Information Processing Systems*, 37:52723–52748.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. 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)*, pages 13484–13508.

Zifeng Wang, Chun-Liang Li, Vincent Perot, Long Le, Jin Miao, Zizhao Zhang, Chen-Yu Lee, and Tomas Pfister. 2024. Codeclm: Aligning language models with tailored synthetic data. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3712–3729.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yifan Wei, Xiaoyan Yu, Ran Song, Hao Peng, and Angsheng Li. 2025. Setke: Knowledge editing for knowledge elements overlap. *arXiv preprint arXiv:2504.20972*.

Yifan Wei, Xiaoyan Yu, Yixuan Weng, Huanhuan Ma, Yuanzhe Zhang, Jun Zhao, and Kang Liu. 2024. Does knowledge localization hold true? surprising differences between entity and relation perspectives in language models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 4118–4122.

Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Weidi Xie, and Yanfeng Wang. 2024. Pmc-llama: toward building open-source language models for medicine. *Journal of the American Medical Informatics Association*, 31(9):1833–1843.

Kai Xiong, Xiao Ding, Li Du, Jiahao Ying, Ting Liu, Bing Qin, and Yixin Cao. 2024. Diagnosing and remedying knowledge deficiencies in llms via label-free curricular meaningful learning. *arXiv preprint arXiv:2408.11431*.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023a. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024b. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Rui Yang, Haoran Liu, Edison Marrese-Taylor, Qingcheng Zeng, Yuhe Ke, Wanxin Li, Lechao Cheng, Qingyu Chen, James Caverlee, Yutaka Matsuo, et al. 2024c. Kg-rank: Enhancing large language models for medical qa with knowledge graphs and ranking techniques. In *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, pages 155–166.

Yuqing Yang, Ethan Chern, Xipeng Qiu, Graham Neubig, and Pengfei Liu. 2023b. Alignment for honesty. *arXiv preprint arXiv:2312.07000*.

Gregory Yauney, Emily Reif, and David Mimno. 2023. Data similarity is not enough to explain language model performance. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11295–11304.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8653–8665, Toronto, Canada. Association for Computational Linguistics.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. 2025. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*.

Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D Goodman. 2024. Quiet-star: Language models can teach themselves to think before speaking. *arXiv preprint arXiv:2403.09629*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi R Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2023. R-tuning: Teaching large language models to refuse unknown questions. *arXiv preprint arXiv:2311.09677*.

Kaiyan Zhang, Ning Ding, Biqing Qi, Sihang Zeng, Haoxin Li, Xuekai Zhu, Zhang-Ren Chen, and Bowen Zhou. 2024. Ultramedical: Building specialized generalists in biomedicine. https://github.com/TsinghuaC3I/UltraMedical.

Hanyu Zhao, Li Du, Yiming Ju, Chengwei Wu, and Tengfei Pan. 2024. Beyond iid: Optimizing instruction learning from the perspective of instruction interaction and dependency. *arXiv preprint arXiv:2409.07045*.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please refer to Abstract Section and Introduction Section 1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to Section B.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Theorems and Lemmas that the proof relies upon have been properly referenced as shown in Section 3.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please refer to the experimental setting 4 and A.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Please refer to the Supplementary Material.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Please refer to the experimental setting 4 and A.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All the results are obtained with fixed random seeds, and we repeat the experiment three times to verify the consistency of the experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please refer to Section D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have understood the NeurIPS Code of Ethics in detail and promise that it has not been violated.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Please refer to Section C.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work does not involve the release of data or models with a high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: All models, data, or codes used in our work are free and open sourced.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide documentation for the use of the code.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our work do not involve crowdsourcing.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our work do not involve this question.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: Our work do not involve this question.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Technical Appendices and Supplementary Material

A.1 Prompts for Synthetic Data Generation Stage

This section introduces the prompts (Figure 5 and 7) defined in our synthetic data generation phase, including the question-answer paris generation prompt, and the evaluation prompt. And Figure 6 shows a specific example generated by SENATOR using the generation prompt.

Synthetic Data Generator (Step 1)

For given facts, generate a question and its corresponding answer. The question should be designed to inquire about the relationship or classification described in the triples, and the answer should be an entity mentioned in the provided facts.
Facts:
Disease <Thyroid Gland Mucoepidermoid Carcinoma> is a type of disease <thyroid gland carcinoma>.
Compound <Liothyronine> treats disease <thyroid gland carcinoma>.
Question: What compound can be used to treat Thyroid Gland Mucoepidermoid Carcinoma?
Answer: Liothyronine.

Facts:
Disease <thyroid gland carcinoma> resembles disease <ganglioneuroma>
Disease <ganglioneuroma> presents Symptom <Diarrhea>
Question: What symptom is associated with the disease that resembles thyroid gland carcinoma?
Answer: Diarrhea.

Facts:
Disease <head and neck cancer> resembles <thyroid gland carcinoma>.
Disease <head and neck cancer> presents Symptom <Dysphonia>.
Disease <head and neck cancer> presents Symptom <Neck Pain>.
Disease <thyroid gland carcinoma> presents Symptom <Dysphonia>.
Disease <thyroid gland carcinoma> presents Symptom <Neck Pain>.
Compound <Paclitaxel> treats disease <head and neck cancer>.
Question: What disease is similar to thyroid gland carcinoma, with Symptom Dysphonia and Neck Pain.
Answer: Head and neck cancer.

Figure 5: Example prompt for the synthetic data generation stage of SENATOR.

A Smaple Generated by SENATOR

{generation prompt}
Input: Maximum Structual Entropy Trajectory by SENATOR
Disease <hyperphosphatemia> contraindicates the use of compound <Retinol>,
Compound <Retinol> is contained in food <hickory nut>,
Food <hickory nut> contains compound <Tryptophan>,
Compound <Tryptophan> is contained in food <cow milk (liquid)>

Output: QA Samples generated by the LLMs
Question: Which compound, present in both hickory nut and cow milk (liquid), is safe for consumption by an individual with hyperphosphatemia?
Answer: Tryptophan.

Figure 6: A specific example generated by SENATOR.

A.2 Prompts for the SFT Evaluation Stage

This section introduces the evaluation prompt (Figure 8) used after model knowledge repair, as shown in Figure 1h, designed to align the model's output answers with the desired format in the medical domain. Specifically, we employ a zero-shot setting in our evaluation to reduce the model's sensitivity bias to few-shot examples.

Sample Evaluation Scorer (Step 2)

Your task is to evaluate the given QA Pairs with Evidences based on the following criteria.
The criteria should include three parts:

Format: Verify the question is complete (i.e., not truncated) and can be answered with a single, clear answer. Check the answer is complete (i.e., not truncated) and is presented in a single entity or a concise subject-predicate-object statement.

Logic: Confirm that there is a clear, derivable logical connection between the question and the answer based on the provided Evidences.

Hallucination: This examines whether the entities involved in the question and answer exist within the provided Evidences. It determines if additional information beyond the given Evidences was used to construct the samples.

For each QA sample, analyze whether it meets the above criteria. If the sample satisfies all criteria, output "Correct". Otherwise, output one of the error types that best describes the issue with the sample: Format, Logic, or Hallucination.

Figure 7: Example prompt for the sample filtering stage of SENATOR.

Eval Prompt for Medical Datasets

 # Instruction:
Directly answer the best option or Directly answer yes/no/maybe:

 # Example (PubMedQA):
Abstract: Electrical neurostimulation has traditionally been limited to the use of charge-balanced waveforms. Charge-imbalanced and monophasic waveforms are not used to deliver clinical therapy, because it is believed that these stimulation paradigms may generate noxious electrochemical species that cause tissue damage. In this study, we investigated the dissolution of platinum as one of such irreversible reactions over a range of charge densities up to 160 μC cm. We observed that platinum dissolution decreased during charge-imbalanced and monophasic stimulation when compared to charge-balanced waveforms
Question: Does electrical neurostimulation with imbalanced waveform mitigate dissolution of platinum electrodes?

 # Example (MedQA)
A 3-month-old baby died suddenly at night while asleep. His mother noticed that he had died only after she awoke in the morning. No cause of death was determined based on the autopsy. Which of the following precautions could have prevented the death of the baby?
A. Placing the infant in a supine position on a firm mattress while sleeping.
B. Keeping the infant covered and maintaining a high room temperature.
C. Application of a device to maintain the sleeping position.
D. Avoiding pacifier use during sleep.

 # Example (MedMCQA):
Which vitamin is supplied from only animal source:
A. Vitamin C B. Vitamin B7 C. Vitamin B12 D. Vitamin D

Figure 8: Example prompt for the evaluation on medical datasets, where the "#" symbol denotes comments illustrating how a specific data sample is combined with an instruction for zero-shot prompting.

A.3 Supervised fine-tuning hyperparameters

We use cross-entropy for supervised fine-tuning. Table 2 presents the hyperparameters utilized for SFT of LLMs within the SENATOR framework. As shown in Table 2, the settings applied to Llama-3-8B are identical to those of Qwen2-7B. Moreover, all experiments conducted in this paper have been performed using the same hyperparameter configuration.

Table 2: Model Training Parameters in SENATOR

Model	Learning Rate	Weight Decay	Warmup Step	Batch Size	Epoch	Maximum Sequence Length
Llama-3-8B	9.65e-6	-1	-1	1	3	1024
Qwen2-7B	9.65e-6	-1	-1	1	3	1024

A.4 Data Filtering

While our framework demonstrates significant improvements over baseline methods, we acknowledge that the system remains imperfect. To systematically evaluate its limitations, we conduct a manual examination of 501 randomly sampled QA pairs from SENATOR outputs. The analysis revealed that 311 samples (62.08%) met our quality criteria for valid question-answer pairs. The remaining 190 error-containing samples (37.92%) exhibited the following error distribution: Formulaic errors (84 samples; 16.77%): Questions or answers with truncations, formatting inconsistencies, or multi-answer requirements. Logical errors (98 samples; 19.56%): Answers lacking evidential support from the provided knowledge triples. Hallucination errors (8 samples; 1.59%): Answers referencing entities absent in the supporting evidence. Notably, while our approach effectively mitigates hallucination errors through evidence grounding, generating logically consistent QA pairs remains challenging. This primarily stems from the base model's inherent limitations in performing multi-hop reasoning across knowledge path. Appendix A.7 illustrates representative examples of these error categories, demonstrating both the framework's capabilities and its current limitations. In order to improve data quality, we set up an additional data filtering module. For format problems, we use regularization to remove samples that do not meet specifications. For logical error types, we use LLMs to judge the logical consistency of QA pairs and evidences, and filter out unsatisfied samples.

A.5 Impact of synthetic data on different medical subfields

Similar phenomena as shown in 4 can also be observed in different medical-related subdomains in the MMLU dataset, as shown in Figure 9. Our analysis on Qwen2 shows that without sythetic data generated by SENATOR (ratio = 0), performance is lowest. As synthetic data increases, sub-domain performance improves but with fluctuations. We attribute this to SENATOR's lack of entity type consideration during KG exploration, causing random data domains and non-uniform categories. Future work will focus on adding entity type constraints in MCTS search to explore domain specific knowledge deficiencies more precisely.

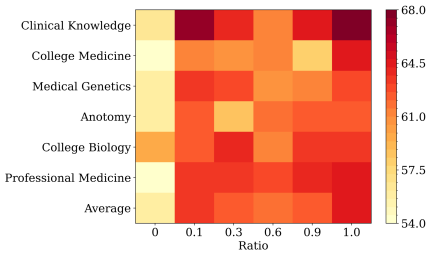


Figure 9: Performance across Different Ratios in MMLU Medical Aspects.

A.6 Comparison with Latest Medical LLM Baselines

To provide a more comprehensive evaluation against recent state-of-the-art medical LLMs, we have added new baselines including BioMistral-7B (Labrak et al., 2024), Meditron-7B (Chen et al., 2023b), Llama-3-8B-UltraMedical (Zhang et al., 2024), and Qwen2-7B w/ SENATOR. The results are presented in Table 3 below.

Table 3: Model Performance on Medical QA Benchmarks

Model	**MedQA**	**MedMCQA**	**PubMedQA**
BioMistral-7B	44.93	42.17	56.4
Meditron-7B	30.40	31.22	61.6
Llama-3-8B-UltraMedical	56.75	53.75	52.12
Llama-3-8B w/ SENATOR	58.29	53.60	64.8
Qwen2-7B w/ SENATOR	59.70	60.70	63.2

A.7 Case Stduy

Our framework SENATOR generates <evidence, question, answer> examples based on the SPOKE knowledge graph. These examples are categorized into four types: Correct, Formulaic errors, Logical errors, and Hallucination errors. Specific examples are illustrated in Figures 10 to 13.

Evidence:
Disease <hyperphosphatemia> contraindicates the use of compound <Retinol>,
Compound <Retinol> is contained in food <hickory nut>,
Food <hickory nut> contains compound <Tryptophan>,
Compound <Tryptophan> is contained in food <cow milk (liquid)>

Question:
What food contains the compound that is contraindicated in hyperphosphatemia?

Answer: Hickory nut

Comment: Correct

Figure 10: Correct Case.

Evidence:
Disease <primary ciliary dyskinesia 25> is a type of disease <primary ciliary dyskinesia>,
In genetics, disease <primary ciliary dyskinesia> associates with gene <MCIDAS>,
Gene <MCIDAS> downregulated in tissue <ectocervix>

Question:
In which tissue is gene MCIDAS upregulated?

Answer: Endometrium

Comment: Hallucination error

Figure 11: Hallucination Error Case.

A.8 Details of the Instruction Tuning Dataset

Medical Conversation Data: the dataset includes approximately 100k instances from the ChatDoctor corpus, which contains diverse doctor-patient dialogues collected from real-world scenarios. To enhance instruction diversity and robustness, each prompt is expanded into multiple semantically equivalent forms using GPT-4.

Medical Rationale Question Answering: the dataset incorporates three major multiple-choice QA benchmarks: MedQA (10.2K examples), MedMCQA (183K), and PubMedQA (211K). These datasets evaluate the model's ability to reason over professional medical knowledge. Since many of these resources originally lacked detailed rationales, additional causal explanations were obtained by prompting ChatGPT, allowing the model to learn both the correct answer and the underlying reasoning.

Evidence:
Disease <acute necrotizing encephalitis> resembles disease <encephalomyelitis>,
Disease <encephalomyelitis> presents symptom <Myalgia>,
Symptom <Myalgia> can be caused by the side effect of compound <Diazepam>

Question:
What disease has a similar presentation to acute necrotizing encephalitis, with a symptom that can be treated by Diazepam?

Answer: Encephalomyelitis

Comment: Logical error

Figure 12: Logical Error Case.

Evidence:
Disease <otulipenia> is a type of disease <autosomal recessive disease>,
Disease <autosomal recessive disease> includes disease <spondyloepiphyseal dysplasia Kondo-Fu type>,
Disease <spondyloepiphyseal dysplasia Kondo-Fu type> presents symptom <Cataract>,
Symptom <Cataract> can be caused by the side effect of compound <Imatinib>

Question:
What is the type of disease that presents symptom Cataract, and what is the side effect of Imatinib?

Answer: Spondyloepiphyseal dysplasia Kondo-Fu type, Cataract

Comment: Formulaic error

Figure 13: Formulaic Error Case.

Knowledge Graph–Driven Prompting: Furthermore, two smaller datasets—LiveQA (635 examples) and MedicationQA (690 examples)—are included to provide real-world clinical questions and drug-related knowledge, respectively. Finally, the dataset includes 99K samples derived from the UMLS medical knowledge graph, covering both entity descriptions and inter-entity relationships. This component is particularly useful for aligning the model with structured biomedical ontologies.

Together, these seven resources offer a diverse and comprehensive instruction set D_I, enabling the model to generalize across conversational, inferential, and knowledge-based medical tasks. More detailed information can be found in the (Wu et al., 2024)

B Limitations

While SENATOR demonstrates promising results in identifying and repairing knowledge deficiencies within LLMs, several limitations remain. First, our framework relies on an external human-curated knowledge graph (KG) to simulate a realistic environment in which the model can perform structured

exploration. This setup enables the LLM to iteratively discover and repair its knowledge gaps through self-improvement. However, such reliance on a high-quality, domain-specific KG may limit the framework's applicability in settings where such structured resources are incomplete or unavailable. In future work, we plan to explore ways to relax this dependency, such as constructing approximate KGs automatically from textual corpora or using retrieval-augmented methods to complement structural guidance.

Second, while the structural entropy-guided exploration effectively identifies knowledge deficiencies, the process of synthesizing data to repair these deficiencies can be further improved. The quality of synthetic data plays a crucial role in downstream model performance. However, this paper places greater emphasis on detecting and targeting knowledge gaps rather than exhaustively optimizing the data generation process. In our current implementation, we adopt prompt-based synthesis strategies for simplicity and reliability. In future work, we aim to incorporate more advanced techniques—such as instruction-tuned generation, controllable sampling to enhance the relevance, diversity, and factuality of the synthesized data.

C Broader Impacts

Our work on the SENATOR framework for detecting and repairing knowledge deficiencies in large language models through targeted synthetic data generation has both promising benefits and potential risks for society.

Positive Impacts

- **Improved Reliability in High-Stakes Domains:** By systematically identifying and closing knowledge gaps, SENATOR can make LLMs more accurate and trustworthy in domains such as medicine, law, and scientific research, where factual precision is critical for patient care, legal reasoning, and scientific discovery.
- **Democratization of Domain-Adapted Models:** Synthetic data alleviates the dependence on expensive, expert-annotated corpora, enabling smaller organizations, research labs, and underserved communities to fine-tune powerful LLMs for specialized tasks without prohibitive annotation costs.
- **Rapid Adaptation to Emerging Knowledge:** In fast-moving fields (e.g., novel pathogens, new regulations), synthetic data guided by up-to-date knowledge graphs can help models stay current, supporting timely decision-making and dissemination of accurate information.

Negative Impacts

- **Bias Amplification and Inaccuracy:** If the underlying knowledge graph or pretraining data contain biases or errors, synthetic data may inadvertently reinforce these issues. Models improved on such data could perpetuate harmful stereotypes or spread misinformation.
- **Misuse for Misinformation:** High-quality synthetic data generation techniques could be exploited to create convincingly false or misleading domain-specific content (e.g., fraudulent medical advice or fabricated legal precedents), posing risks to public trust and safety.
- **Overreliance on Synthetic Data:** An overconfidence in models fine-tuned primarily on synthetic data might obscure residual blind spots, leading users to place undue trust in automated systems without appropriate human oversight.
- **Privacy and Intellectual Property Concerns:** If knowledge graphs incorporate sensitive or proprietary information, there is potential for synthetic data to leak or replicate protected content, raising ethical and legal implications.

D Resource Requirement

We use 8 NVIDIA A100-40G GPUs to SFT Llama-3-8B and Qwen2-7B, and leverage 1-2 NVIDIA A100-40G GPUs for all the inference experiments.

Taking Qwen2-7B as an example, when using synthetic data to SFT Qwen2-7B for knowledge repair, the training time is about 30h on 8 NVIDIA A100-40G GPUs, and a total of 3 epochs are performed.

The inference time such as synthetic data generation stage and evaluation stage, measured in seconds per sample, is calculated on an NVIDIA A100 GPU with vllm acceleration (e.g. Qwen2-7B model, which demands at least two A100 GPUs for deployment)