Human-Pedagogy Inspired LLM Fine-Tuning Paradigm for Lifelong Leaning and Continual Adaptation

Nitin Vetcha[1,2,*]

[1]Department of Computational and Data Sciences, Indian Institute of Science, Bangalore, Karnataka, India
[2]Department of Ophthalmology, Yong Loo Lin School of Medicine, National University of Singapore, Singapore

Abstract

Current Large Language Model (LLM) training paradigms, while effective at pattern matching and knowledge retrieval, often fall short of replicating the nuanced, adaptive and generalizable reasoning characteristics of human intelligence. We argue that this stems from a fundamental disconnect between the static, data-driven training of LLMs and the dynamic, lifelong learning process inherent to human cognitive development that naturally resolves the *stability-plasticity dilemma* arising while deploying LLMs in non-stationary environments. Traditional approaches like experience replay or regularization often treat data as static points, ignoring the cognitive structures of learning. To bridge this gap, we introduce a novel robust blueprint for streaming and continual learning (SCL), namely Learn-Master-Teach Tuning (LMT^2) , a visionary end-to-end training framework that simulates the complete human 'student-to-teacher' life-cycle. Our paradigm guides the model through a comprehensive developmental trajectory, from a novice learner internalizing a curriculum to a seasoned educator capable of lifelong learning and knowledge synthesis. By situating learning within a holistic, cognitive-inspired framework, we explore two fundamental research questions: Can the deep simulation of a human persona, in this case, a developing academic, act as a proxy for drift detection and active learning in streaming environments? And, does this student-teacher life-cycle offer a superior training paradigm to resolve the plasticity-stability dilemma compared to traditional continual fine-tuning in LLMs? We present the complete LMT^2 methodology and position it within the landscape of existing SCL training paradigms, arguing that by emulating the human journey of learning, we can unlock new frontiers thereby enabling LLMs to operate in dynamic, streaming data environments.

Keywords

Lifelong Learning, Catastrophic Forgetting, Large Language Model Fine-tuning, Human Pedagogy

1. Introduction

LLMs have demonstrated remarkable capabilities in a wide range of downstream natural language tasks, largely due to their ability to learn from vast amounts of text data and development life-cycle, which typically unfolds across three canonical stages: vast, self-supervised pre-training on web-scale text corpora; supervised fine-tuning (SFT) on curated instruction-response pairs; and alignment through reinforcement learning, often with human feedback (RLHF). This approach, despite resulting in remarkable success, treats learning as a process of mass data ingestion, focused on statistical pattern recognition rather than a structured, developmental journey. The resulting models, while possessing broad knowledge, are merely "approximate omniscients" that excel at next-token prediction but lack the deep, verifiable expertise and robust reasoning due to de-contextualized learning. This stands in stark contrast to the robust, flexible, and continuously evolving nature of human cognition which is what is truly necessary in in real-world applications as data distributions shift, new vocabulary emerges and factual knowledge evolves.

Streaming Continual Learning (SCL) aims to address this by updating models on the fly. However, current LLM fine-tuning methods, such as LoRA or full-parameter tuning, are susceptible to *catastrophic forgetting* where updating weights for new data, say D_{t+1} destroys the representations learned for

1st Streaming Continual Learning Bridge at AAAI26, January 21, 2026, Singapore.
*Corresponding author.
nitinvetcha@iisc.ac.in (N. Vetcha)
https://github.com/nitinvetcha/ (N. Vetcha)
0009-0003-6542-324X (N. Vetcha)

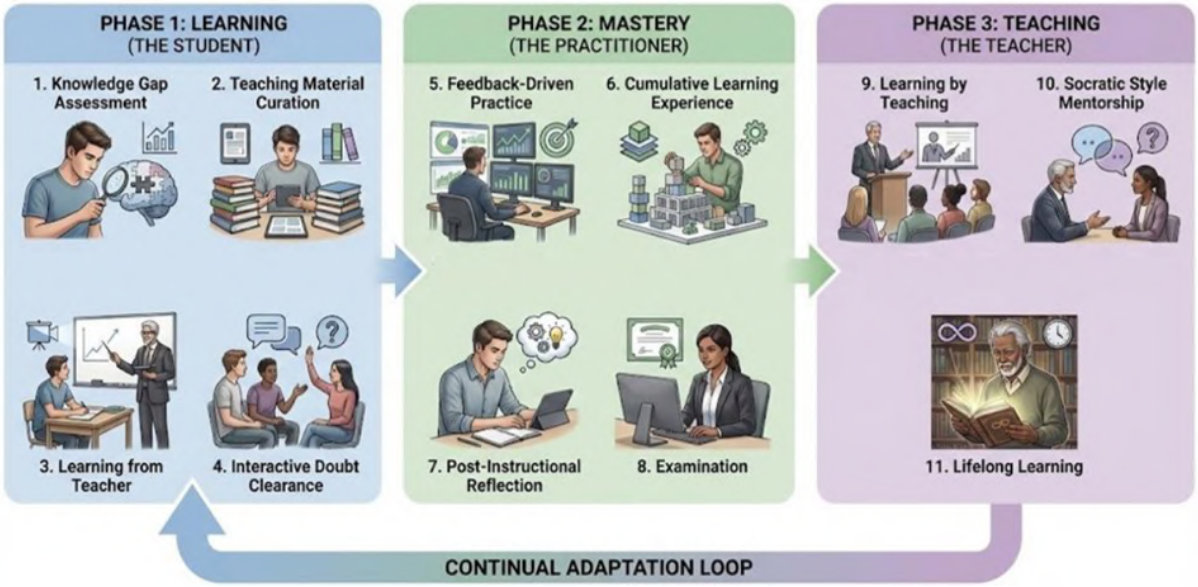

Figure 1: *Conceptual vision of the proposed Learn-Master-Teach (LMT^2) Tuning framework* - The paradigm simulates the cognitive developmental life-cycle of a human expert to achieve lifelong learning. It progresses through three phases: (1) Learning (The Student), focusing on knowledge acquisition and gap assessment; (2) Mastery (The Practitioner), focusing on feedback-driven application and refinement; and (3) Teaching (The Teacher), focusing on knowledge consolidation and sharing. This structured curriculum enables the model to continuously adapt to new information while retaining prior capabilities.

data at a previous timestep, say D_t. Existing SCL solutions typically fall into three categories: (1) *regularization-based* (e.g., elastic weight consolidation [1]), which constrain weight updates; (2) *replay-based* (ex: [2]) which store old data and (3) *architecture-based* (ex: [3]), which expand model capacity. While effective for classification, these methods often fail to capture the semantic nuance of language tasks, treating all tokens as equal and all updates as equally valid. We posit that the solution lies in moving beyond simple regularization techniques and towards a holistic training paradigm inspired by human pedagogy. Humans do not learn by simply appending new data to a mental buffer. We operate in a structured manner which can be broadly seen as comprising of three phases (see Figure 1),

- **Phase 1: Learning (The Student)** corresponds to *Plasticity*. It involves detecting knowledge gaps in the stream and curating curriculum to address them without overfitting to noise.
- **Phase 2: Mastery (The Practitioner)** corresponds to *Robustness*. It involves applying knowledge through feedback loops and self-correction to ensure the model isn't just memorizing the stream but generalizing from it.
- **Phase 3: Teaching (The Teacher)** corresponds to *Stability*. It involves consolidating knowledge into long-term memory and using "generative replay" (teaching) to reinforce old concepts.

We therefore aim to translate these pedagogical stages into concrete machine learning mechanisms suitable for SCL and address the following research questions,

RQ 1: Can the deep simulation of a human persona in LLMs, which in this case is of a developing academic, translate to replication of the persona's capability thereby acting as a proxy for drift detection and active learning in streaming environments?

RQ 2: Does a student-teacher life-cycle offer a superior training paradigm with for LLMs to resolve the plasticity-stability dilemma thereby achieving superior retention and adaptation on non-stationary data streams, compared to traditional continual FT?

The major contributions of this paper include

- LMT^2 (Learn-Master-Teach Tuning), a novel multi-stage training paradigm with an end-to-end framework that instead of treating the model as a static entity to be filled with information, guides it through a complete, simulated human life-cycle of learning and growth, from a student to a teacher (see Fig. 1)
- MentorX, a 7B-LMT tuned model trained to be an adaptive educational tutor for K-12 mathematics which addresses RQ 1 and SkolarX, which is another 7B-LMT tuned model achieving comparable performance with SFT baselines with significantly lower training data as a response to RQ 2

Rest of the paper is organized as follows: Section 2 discusses the motivation behind our approach, Section 3 constitutes the relevant literature survey followed by the proposed LMT^2 methodology in Section 4 and the corresponding experiments in Section 5. Sections 6, 7 and 8 present the limitations, future research directions and conclusions.

2. Motivation

Our primary motivation is to explore the transformative potential of **situated learning** and **cognitive apprenticeship** in the context of SCL for LLMs. Lave and Wenger's theory of situated learning [4] posits that learning is not the mere transmission of abstract knowledge, but an integral part of social practice. Similarly, Collins, Brown, and Newman's model of cognitive apprenticeship [5] emphasizes the importance of learning in the context of authentic activity, with expert guidance and modeling. LMT^2 is designed to be a computational instantiation of these theories in the SCL paradigm, providing a simulated "social practice" and "authentic activity" for LLM's lifelong learning with continual adaptation.

This leads to our first core motivation, encapsulated in RQ 1. While LLMs are adept at persona imitation, it is unclear if this is a shallow form of mimicry or if it can lead to a deeper embodiment of a persona's traits and abilities. Psychological literature suggests that identity formation is deeply intertwined with lived experience. By having our "MentorX" agent progress through the distinct stages of a student-teacher life-cycle, we aim to investigate whether this simulated "lived experience" can foster a more genuine form of intelligence capable of SCL, *one that is not just knowledgeable about a domain, but can reason and act within it.*

Our second motivation, addressed by RQ 2, is the pursuit of a more effective training SCL paradigm. The student-teacher life-cycle is a powerful engine for learning in humans. The process of learning, being tested, and then having to teach others forces a deeper understanding of the material, promotes self-reflection and encourages the development of more robust mental models while keeping prior information intact. We hypothesize that an LLM that undergoes this same process will develop more generalizable reasoning skills, better error-correction abilities, and a greater capacity for lifelong learning. This is a departure from the current paradigm, which often results in models that are "a mile wide and an inch deep."

3. Related Works

Streaming Continual Learning: For LLMs, SCL has emerged as a critical research area to address the challenge of updating pre-trained models on non-stationary data streams while mitigating catastrophic forgetting, a central issue for deployment in dynamic environments. Recent comprehensive surveys have highlighted multi-stage categorization schemes like continual pre-training, continual instruction tuning, and continual alignment as foundational mechanisms for lifelong adaptation and retention of knowledge over time [2], as well as broader taxonomies of internal and external lifelong learning strategies [3]. Traditional CL research borrows extensively from classical mechanisms like regularization, replay, parameter-efficient adaptation and modular expansion so as to balance plasticity

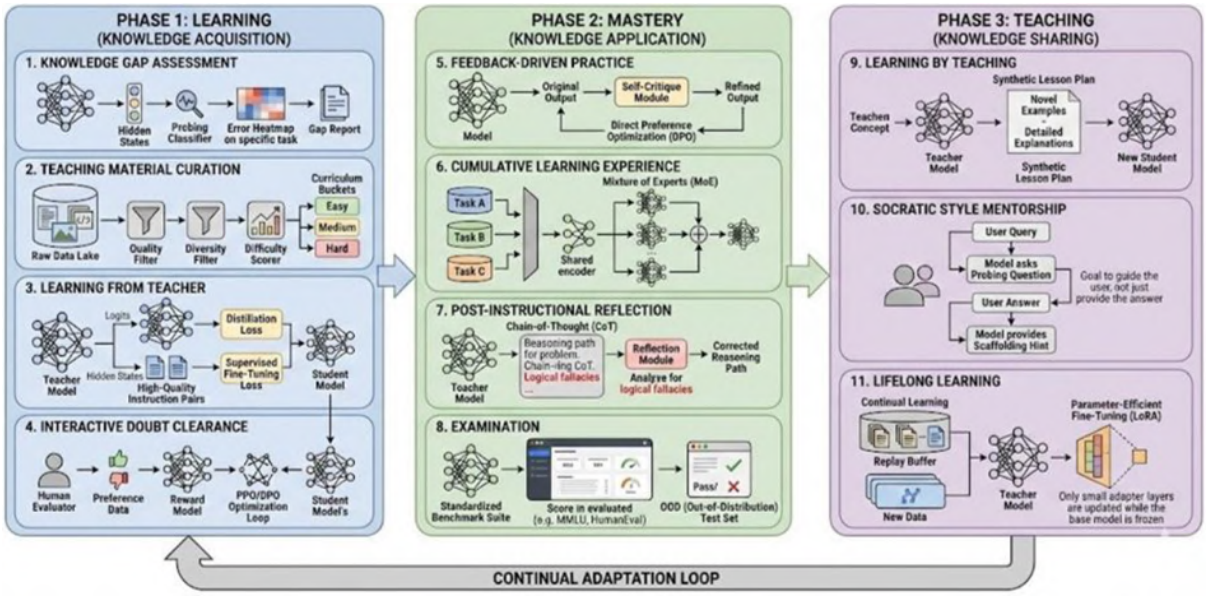

Figure 2: *Detailed system implementation of the LMT^2 pipeline* - The framework integrates specific modular components for each stage of the continual adaptation loop. Key components include Knowledge Gap Assessment using probing classifiers to detect drift, Feedback-Driven Practice utilizing Direct Preference Optimization (DPO) for refinement, Cumulative Learning via Mixture of Experts (MoE) to manage task interference and Lifelong Learning using WISE to efficiently update the model on new data streams while freezing the base model.

and stability, but these often treat streaming updates as algorithmic fixes rather than structured developmental processes. Moreover, most existing approaches focus on algorithmic mitigation of forgetting or evaluation benchmarks for sequential tasks, with limited work framing the continual adaptation as a developmental curriculum that mirrors human lifelong learning. In contrast, the proposed LMT^2 paradigm introduces a holistic, human-pedagogy inspired training pipeline that aligns stages of learning, mastery, and generative teaching with core SCL objectives, thereby embedding curriculum structuring, meta-reflection and generative replay into the continual learning process itself. By situating the training within this cognitive-inspired sequence, LMT^2 complements existing SCL strategies with a structured pedagogical lens rooted in curriculum theory and lifelong learning principles.

LLM-Based Data Augmentation: Data augmentation techniques for LLM post-training have become essential for improving model performance, especially when labeled data is scarce or diverse data is needed. Common strategies include prompt-based augmentation, where LLMs generate new training examples by rephrasing, paraphrasing, or expanding existing data using carefully designed prompts, and retrieval-based augmentation, which incorporates external knowledge to produce more grounded and contextually rich data. Hybrid approaches combine these methods to maximize both diversity and faithfulness of the generated samples.

Multi-Stage LLM Post-Training Paradigms: Multi-stage post-training paradigms for LLMs are emerging as powerful strategies to enhance model capabilities, generalization, and alignment with complex tasks. These approaches often involve sequential or joint fine-tuning steps, such as supervised fine-tuning (SFT) followed by preference learning (e.g., RLHF or DPO), or modular training where different components of a system are specialized and refined in stages. Recent research highlights the limitations of simple sequential post-training, showing that models can “forget" earlier training stages, and proposes joint or co-training frameworks to mitigate this issue and improve overall performance. Multi-agent and multi-component paradigms, where several LLMs or modules collaborate and are trained together (sometimes with reinforcement learning), have demonstrated superior results in tasks

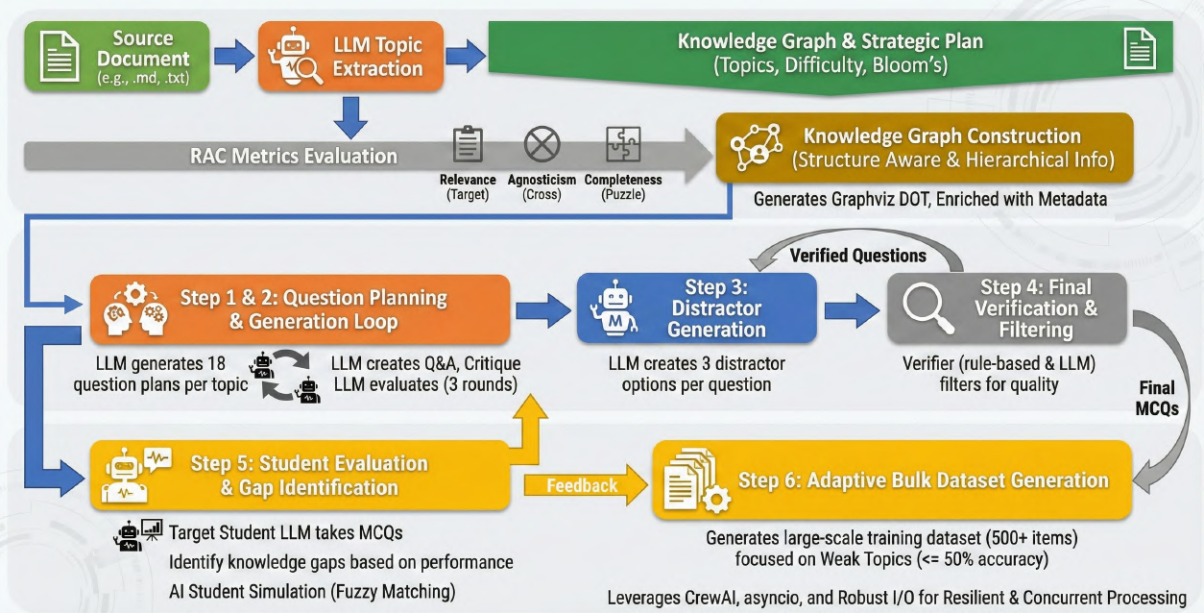

Figure 3: Details of the Stream Structuring via Material Curation Stage and Drift Detection via Knowledge Gap Assessment

requiring reasoning, tool use, or multimodal understanding [6]. Additionally, multi-stage influence functions and progressive enhancement strategies allow for more interpretable and effective adaptation of LLMs to downstream tasks, such as text ranking or retrieval. The use of synthetic data generated through multi-agent simulations further enriches post-training, enabling models to better follow human instructions and generalize to new domains. Overall, multi-stage post-training paradigms represent a shift from static, single-step fine-tuning to dynamic, collaborative, and modular learning processes that unlock new levels of LLM performance and flexibility.

Human-Pedagogy Inspired LLM Enhancement: Recent research on human-pedagogy inspired enhancements for LLMs explores methods such as structured curriculum training [7], iterative teacher-student refinement [8], self-reflection [9] and meta-cognitive strategies to improve educational outcomes. Techniques like simulating teacher-student interactions and generating teaching reflections allow LLMs to iteratively refine teaching plans, achieving quality comparable to those crafted by expert educators and supporting pre-class rehearsal and introspection in lesson design. Fine-tuning LLMs with datasets that emphasize Socratic guidance [10] and conceptual scaffolding, rather than direct answers, leads to more pedagogically aligned models that foster deeper learning and reduce over-assistance, though sometimes at a slight cost to accuracy. Learning from human preferences and synthetic data generation further enhances LLMs' ability to provide scaffolded guidance, supporting meta-cognitive and reflective learning processes. Studies also show that LLMs can automate the creation of open-ended, curiosity-driven question prompts, which help students develop critical thinking and inquiry skills, and that playful, game-based approaches can nurture domain expertise and self-regulation. Human-in-the-loop frameworks, where educators iteratively refine LLM-generated content, lower cognitive demand and increase productivity in instructional design. Additionally, LLMs can personalize pedagogy by recommending best practices and adapting content to students' cultural backgrounds, supporting both introspective teaching and culturally relevant pedagogy. Overall, integrating human-pedagogy principles into LLM training and deployment holds promise for fostering more effective, reflective, and adaptive educational experiences.

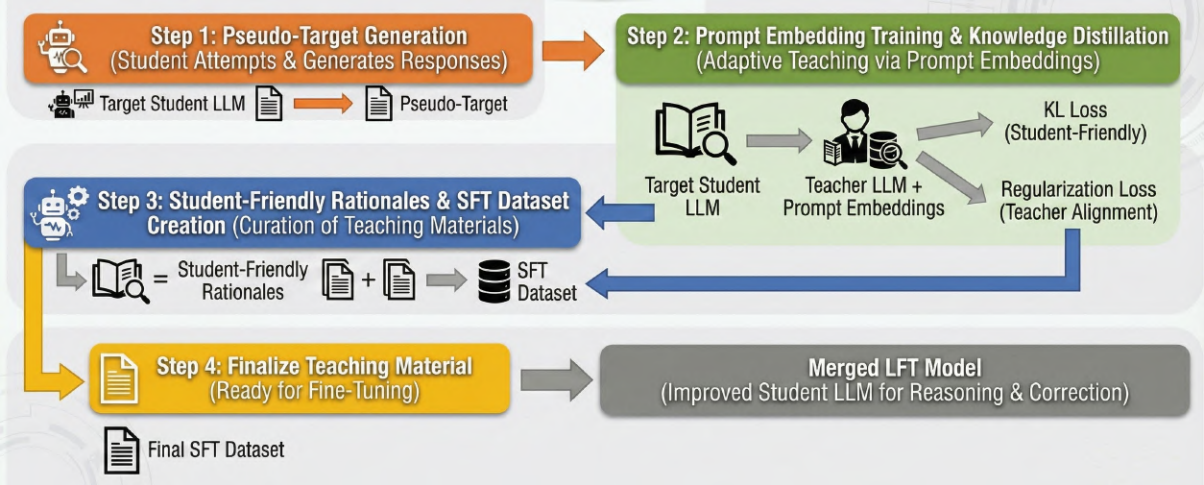

Figure 4: Details of the Structured Injection via Prompt Distillation Stage

4. Methodology

The LMT[2] framework operates as a continuous life-cycle. In the context of Streaming Continual Learning (SCL), we now formalize in detail the three phases i.e., Learning, Mastery, and Teaching as distinct mechanisms to balance the stability-plasticity dilemma.

4.1. Phase 1: Learning (Knowledge Acquisition via Active Streaming)

In a non-stationary environment, a model cannot simply consume all incoming data D_{stream} indiscriminately; doing so leads to catastrophic interference and inefficiency. The "Student" phase of our framework emulates a human learner's ability to structure incoming information, selectively attend to novel concepts, and actively resolve ambiguities. We formalize this as a four-step pipeline: Stream Structuring, Drift-Aware Gap Assessment, Structured Injection, and Active Querying.

4.1.1. Stream Structuring via Material Curation:

Raw data streams are often noisy and unstructured. A human student does not memorize raw text; they convert it into structured mental models (notes, Q&A). To replicate this, we employ a **Stream Structuring Module** utilizing the SciQAG framework [11]. Given an incoming document batch B_t from the stream, instead of performing standard causal language modeling, we transform B_t into a structured set of scientific Question-Answer pairs, $S_t = \{(q, a)_i\}$.

$$S_t = \text{SciQAG}(B_t) \tag{1}$$

This transformation serves two SCL purposes:

1. **Noise Reduction:** By extracting only salient scientific facts into Q&A format, we filter out stylistic noise and irrelevant tokens that contribute to overfitting.
2. **Task Formatting:** It converts unsupervised stream data into an instruction-tuning format, preparing the model for the subsequent active learning step.

4.1.2. Drift Detection via Knowledge Gap Assessment:

A core challenge in SCL is detecting when the data distribution has shifted (concept drift) or when the model lacks specific knowledge (epistemic uncertainty). We model the "Student's" testing phase as a proxy for **active learning**. Before updating weights, we assess the current model M_{θ_t} against the structured stream S_t. We utilize the Knowledge-Aware Fine-Tuning (KaFT) protocol [12] to compute

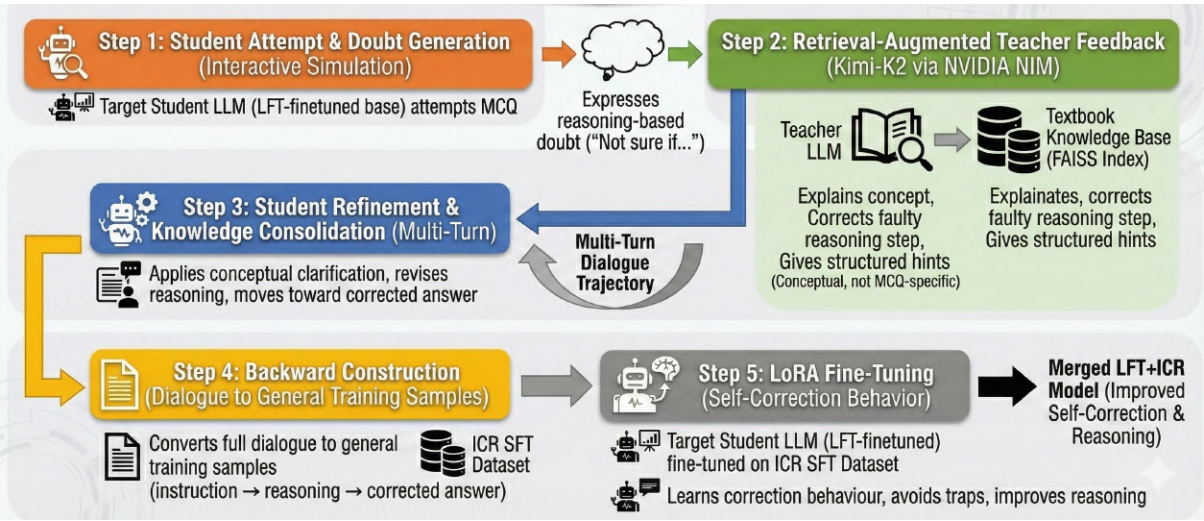

Figure 5: Details of Active Querying via Interactive Doubt Clearance Stage

a conflict score $\mathcal{C}$ for each sample. The model attempts to answer $q_i \in S_t$. If the model's internal knowledge conflicts with the stream data (indicating either a hallucination or an outdated fact due to drift), we flag this sample for high-priority learning.

$$w_i = \mathbb{I}(M_{\theta_t}(q_i) \neq a_i) \cdot \beta + \mathbb{I}(M_{\theta_t}(q_i) \approx a_i) \cdot \gamma \tag{2}$$

where w_i is the sample weight, and $\beta \gg \gamma$. This mechanism acts as a filter for *plasticity* since the model allocates gradient updates primarily to "unknowns" (new concepts or drifts) while suppressing updates for "knowns," thereby naturally mitigating forgetting by reducing unnecessary weight perturbations on established knowledge.

4.1.3. Structured Injection via Prompt Distillation:

Once high-drift samples are identified, we must inject this knowledge without destabilizing existing representations. We employ a **Teacher-Student Distillation** approach for the update step [13, 14]. Instead of raw SFT, we utilize prompt distillation. A frozen, larger teacher model (representing an oracle or a more capable past snapshot) receives the new knowledge k in its context window and generates a reasoning trace. The student model M_θ is optimized to mimic this output distribution.

$$\mathcal{L}_{update} = \sum_{(q,a) \in S_t} w_i \cdot \mathrm{KL}(P_{Teacher}(y|q,k) || P_{Student}(y|q)) \tag{3}$$

This acts as a regularizer. By distilling the distribution rather than fitting hard labels, we smooth the loss landscape, allowing the model to adapt to the stream (high plasticity for high w_i samples) while retaining the structural reasoning capabilities of the teacher.

4.1.4. Active Querying via Interactive Doubt Clearance:

Passive learning from a teacher is often insufficient for resolving deep ambiguities or complex drifts. To address this, we introduce an **Active Querying Module** inspired by the INTERACT framework [15]. When the student model M_θ encounters high uncertainty (high entropy H) in its predictions even after initial injection, it does not passively accept the loss. Instead, it enters an interactive loop . The student generates a clarifying question $q_{clarify}$ targeting the ambiguity, and the Teacher model provides a specific explanation $e_{response}$.

$$\text{If } H(M_\theta(q|k)) > \tau, \quad \text{Query: } q_{clarify} \leftarrow M_\theta(q,k) \tag{4}$$

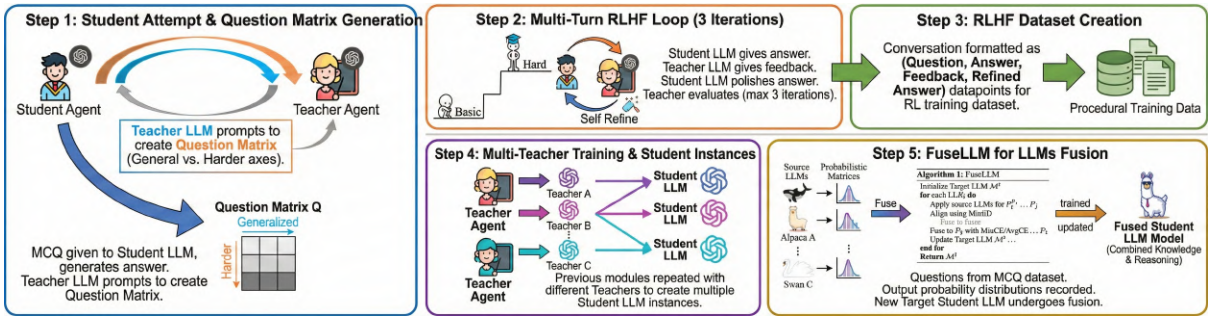

Figure 6: Details of the Active Refinement via Feedback-Driven Practice Stage and Forgetting Mitigation via Cumulative Ensemble Distillation Stage

The student then integrates this response into its context and re-attempts the task. This multi-turn interaction allows the model to refine its internal representations of complex concepts before weight updates occur, effectively reducing epistemic uncertainty and ensuring that only high-confidence, verified knowledge is encoded into long-term memory. This completes the “Student" phase in which the model has structured the stream, identified drift, learned via distillation and actively resolved ambiguities.

4.2. Phase 2: Mastery (Robustness via Recursive Refinement)

While Phase 1 handles the initial acquisition of new stream data, SCL requires that these updates be robust to noise and compatible with prior knowledge. The “Mastery" phase formalizes this as a recursive self-improvement loop, transforming the model from a passive learner into an active practitioner that validates and consolidates new information.

4.2.1. Active Refinement via Feedback-Driven Practice:

In a streaming setting, single-pass training on noisy data often leads to shallow minima. To enforce robustness, we implement an **Iterative Refinement Loop** inspired by the YODA framework [8]. For high-loss samples identified in Phase 1, the model does not merely minimize cross-entropy. Instead, it enters a feedback loop where a teacher agent evaluates the student’s output y_t and provides a critique c_t. The student then generates a refined output y_{t+1} conditioned on this critique.

$$y_{t+1} = M_\theta(x|y_t, c_t) \tag{5}$$

We update the model weights only on the final, verified trajectory (x, y_{final}). This effectively filters out stochastic noise from the stream, ensuring that gradients are computed based on high-confidence, reasoned paths rather than initial, potentially erroneous guesses.

4.2.2. Forgetting Mitigation via Cumulative Ensemble Distillation:

A primary failure mode in SCL is catastrophic forgetting, where fitting the current stream D_t erases knowledge from D_{t-1}. To mitigate this, we employ a **Multi-Teacher Distillation** strategy [16]. Instead of distilling from a single oracle, the student learns from an ensemble of teachers $\mathcal{T} = \{T_{current}, T_{past_1}, T_{past_2}\}$. These represent snapshots of the model at different timesteps or specialized expert models. The update objective minimizes the divergence from the ensemble average, acting as a form of generative replay without storing raw data.

$$\mathcal{L}_{ensemble} = \sum_{T \in \mathcal{T}} \alpha_T \cdot \mathrm{KL}(P_T(y|x)||P_{Student}(y|x)) \tag{6}$$

This forces the model to find a parameter configuration that satisfies both the current stream (via $T_{current}$) and historical constraints (via T_{past}), explicitly balancing plasticity and stability.

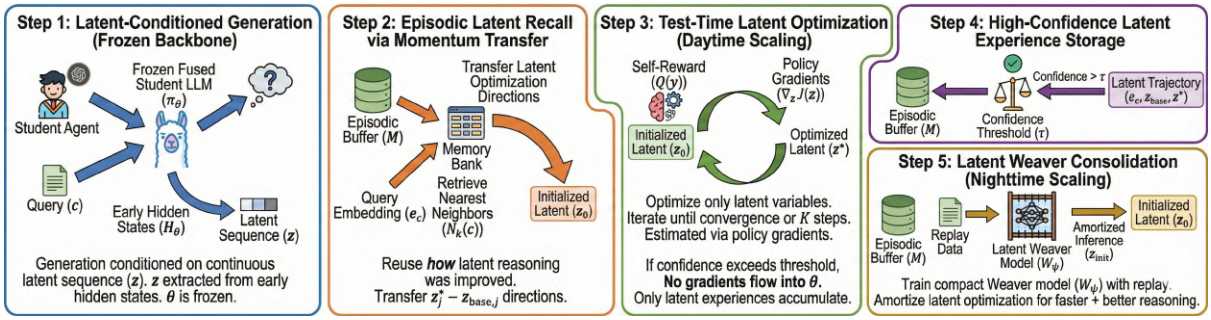

Figure 7: Details of the Drift Adaptation via Post-Instructional Reflection Stage

4.2.3. Drift Adaptation via Post-Instructional Reflection:

When a distribution shift occurs, simply updating weights can lead to incoherent internal representations. To ensure the model understands the drift, we utilize a Meta-Introspection Module such as ReflectEvo [9]. After an update step, the model is prompted to generate a self-reflection r analyzing its own reasoning process on the new data: "Why was my initial prediction wrong? What concept changed?" This generated reflection r is added to the context buffer for future samples in the same stream batch.

$$M_{\theta_{t+1}}(x_{new}) \leftarrow M_{\theta_t}(x_{new}, r) \tag{7}$$

This mechanism acts as an internal regularizer, forcing the model to explicitly verbalize the concept drift, which aids in rapid few-shot adaptation to the new distribution.

4.2.4. Reliability Verification via Peer-Review Examination:

In SCL, it is critical to verify that an update has not degraded performance on previous tasks. We implement a **Peer-Review Gate** using the FAIR approach [17]. Before committing the new weights θ_{new}, a committee of frozen peers evaluates the model on a small anchor set of historical samples. The update is accepted only if the examination score (performance stability) remains above a threshold τ.

$$\text{Update } \theta \leftarrow \theta_{new} \iff \text{Score}_{Peer}(\theta_{new}) \geq \tau \tag{8}$$

This step prevents polluted or destabilizing updates from corrupting the long-term memory, serving as a final quality check in the streaming pipeline.

4.3. Phase 3: Teaching (Stability via Generative Consolidation)

In the final phase of the SCL loop, the model transitions from a consumer of the stream to a generator. This "Teaching" phase is critical for stability; by forcing the model to articulate and restructure its knowledge for a student, we implement a form of *generative replay* and *modular editing* that solidifies long-term retention against non-stationary drift.

4.3.1. Stability via Learning by Teaching (Generative Replay):

To prevent catastrophic forgetting of previous stream concepts, we utilize a *Learning by Teaching* (LbT) paradigm [18]. Instead of simply minimizing loss on the current batch, the model M_θ acts as a "Teacher" and generates synthetic instructional data D_{synth} (rationales, examples) for a weaker student model S_ϕ. The teacher optimizes its own representations to maximize the student's learning efficiency.

$$\mathcal{L}_{LbT} = -\mathbb{E}_{(x,y)\in D_{synth}}[\log P_{S_\phi}(y|x, \text{Rationale}_\theta)] \tag{9}$$

This process forces the Teacher model to generate high-fidelity, generalized representations of the data distribution. By teaching the student, the model effectively replays its internal knowledge, reinforcing its own weights against drift without needing to store the original raw data stream.

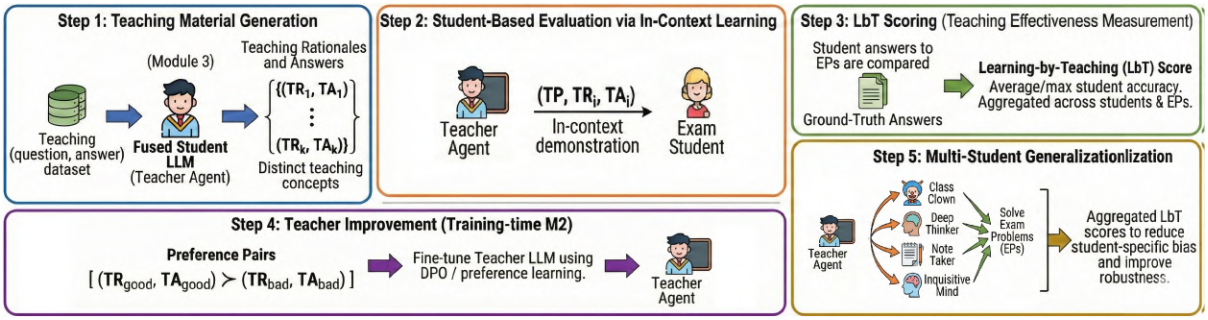

Figure 8: Details of Stability via Learning by Teaching (Generative Replay) Stage

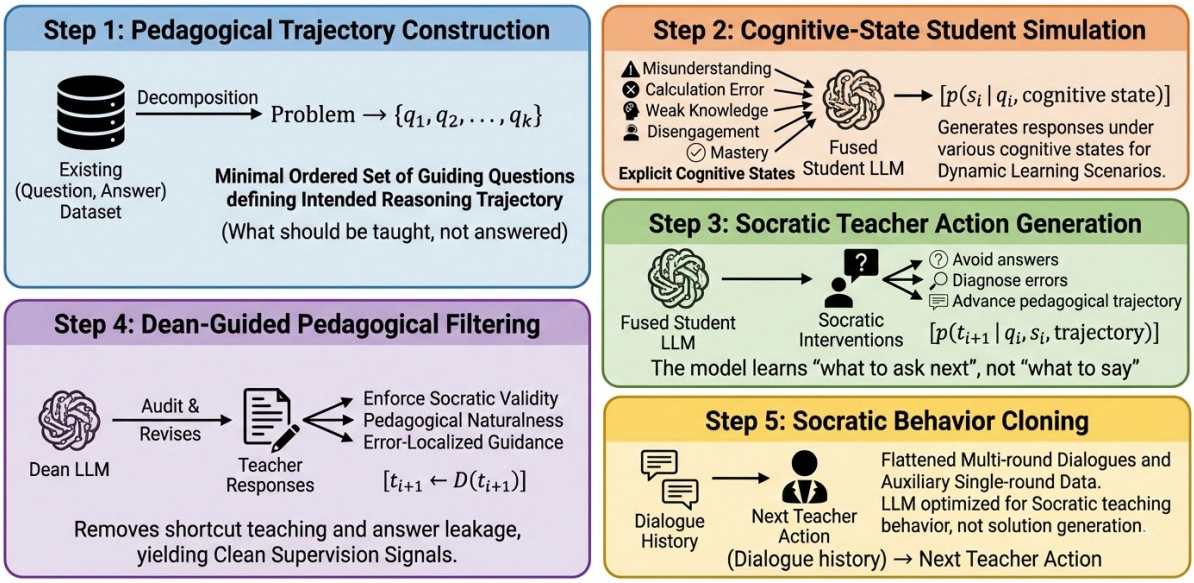

Figure 9: Details of the Policy Optimization via Socratic Mentoring Stage

4.3.2. Policy Optimization via Socratic Mentoring:

Merely outputting answers is insufficient for robust generalization. To ensure the model has internalized the causal structure of the stream data, we train it to act as a Socratic Tutor [19, 10]. We formulate this as a Reinforcement Learning (RL) problem where the model learns a policy π_θ to guide a student through a multi-turn reasoning process. The reward signal R is derived from the student's successful convergence to the correct answer without being given the solution directly.

$$J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} \gamma^t R(s_t, a_t) \right] \tag{10}$$

This optimization ensures that the model learns the underlying logic and dependencies of the domain, rather than just surface-level correlations, making it more robust to adversarial shifts in the data stream.

4.3.3. Non-Stationary Adaptation via Lifelong Memory Editing:

Finally, to handle the "Plasticity-Stability" dilemma in a perpetually non-stationary environment, we employ the WISE Framework** (Working & Side Memory Editing) [20]. We decouple the model's memory into two components:

1. **Main Memory (Θ_{Main}):** Frozen or slowly updating parameters containing general reasoning capabilities (Stability).

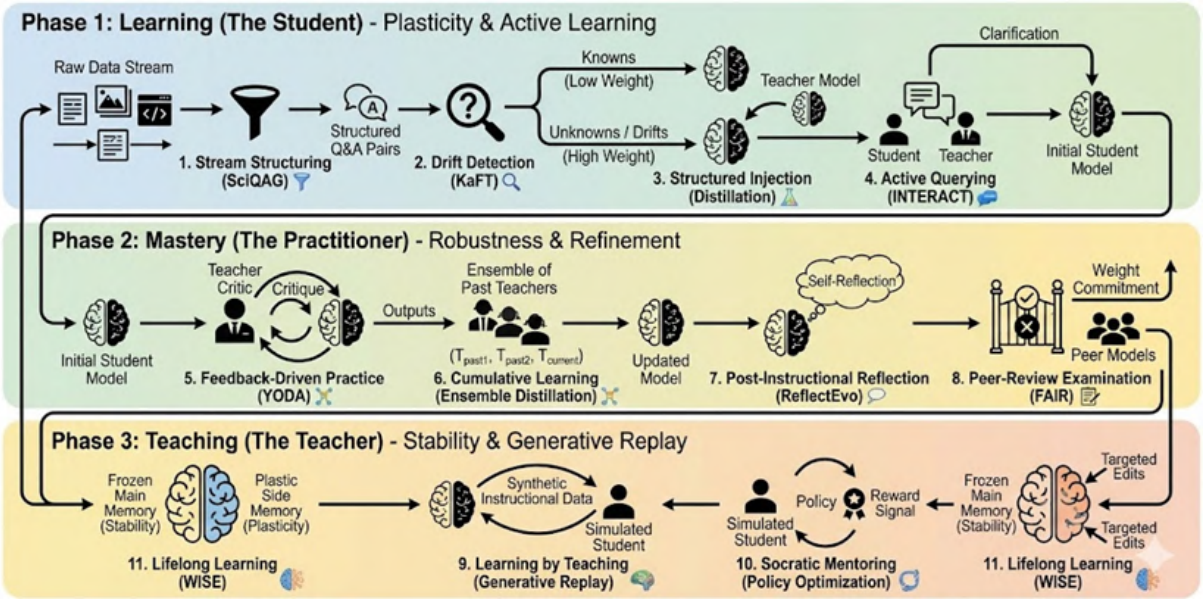

Figure 10: *Technical architecture of LMT^2 mapped to SCL objectives* - The framework addresses the stability-plasticity dilemma through three mechanistic phases: Phase 1 ensures Plasticity via stream structuring and drift detection ; Phase 2 ensures Robustness via recursive feedback and ensemble distillation; and Phase 3 ensures Stability via generative replay and targeted memory editing . This pipeline allows the model to actively learn from non-stationary streams without catastrophic forgetting.

2. **Side Memory (Θ_{Side}):** Rapidly updating adapter modules that store specific, time-sensitive facts from the stream (Plasticity).

A routing mechanism $g(x)$ determines which memory to access for a given query x.

$$y = g(x) \cdot M(\Theta_{Side}; x) + (1 - g(x)) \cdot M(\Theta_{Main}; x) \tag{11}$$

When the stream introduces a factual update (e.g., "The Prime Minister has changed"), we edit only the relevant shard in Θ_{Side}. This allows for precise, localized updates to handle concept drift without catastrophic interference with the global model, enabling true lifelong learning.

5. Experiments

We applied LMT^2 to Llama-3.2-1B-Instruct for the topic of differential equations. The seed documents uploaded include just a single chapter, titled “Differential Equations” from a K-12 mathematics textbook. The evaluation dataset consisted of the corresponding questions from the Big-Math dataset [21], thereby avoiding the possibility of data contamination. The resulting LMT^2 tuned model outperformed the base model by a significant margin of 33% indicating the potential of our pipeline. Due to the generic nature of LMT^2, it can be generalized to domains apart from mathematics as well.

6. Conclusions

In this paper, we have introduced LMT^2, a novel training paradigm that represents a fundamental departure from current approaches to developing LLMs. We have argued that the path to more robust, generalizable, and adaptive AI lies not in piecemeal, modular enhancements, but in redesigning the training process itself to be more holistic and developmental. By proposing a framework that simulates the complete human ’student-to-teacher’ life-cycle, we have provided a concrete methodology for exploring two of the most critical questions in AI research: whether deep simulation can lead to genuine replication of a persona’s capabilities, and whether a human-centric developmental journey constitutes

a superior training paradigm. The framework, with its 3 distinct phases of “In the Classroom," “Mastery" and “The Teacher", is not merely a collection of techniques, but an integrated, cognitive-inspired narrative. It is our belief that by building models that learn in a way that is more analogous to our own journey of intellectual growth, we can begin to bridge the gap between the brittle intelligence of current systems and the fluid, adaptable intelligence that remains the hallmark of the human mind.

7. Acknowledgments

The author would like to thank Professor Sashikumaar Ganesan, from the Department of Computational and Data Science at Indian Institute of Science, Bangalore for providing valuable insightful feedback and the adequate compute resources required to execute this project.

Declaration on Generative AI

During the preparation of this work, the author used Large Language Models (GPT-5.2, Claude Opus 4.5 and Gemini-3) as a writing assistant tool for drafting content, to generate literature review, for abstract drafting, to paraphrase and reword, to improve writing style, for grammar and spelling check as well as to generate the images used in the paper. The process was interactive. After writing the core content, the author used LLMs with specific prompts to refine the text. These prompts included requests to “check for grammatical errors,” “rephrase this sentence for clarity,” “make this paragraph more concise,” or “suggest alternative phrasing to improve flow.” The LLMs were not used to generate any scientific ideas, experimental results, data analysis or other core intellectual contributions of the paper. After using these tool(s)/service(s), the author reviewed and edited the content as needed and takes full responsibility for the publication’s content.

References

[1] V. Šliogeris, P. Daniušis, A. Nakvosas, Elastic weight consolidation for full-parameter continual pre-training of gemma2, arXiv preprint arXiv:2505.05946 (2025).

[2] H. Shi, Z. Xu, H. Wang, W. Qin, W. Wang, Y. Wang, Z. Wang, S. Ebrahimi, H. Wang, Continual learning of large language models: A comprehensive survey, ACM Computing Surveys (2024).

[3] J. Zheng, S. Qiu, C. Shi, Q. Ma, Towards lifelong learning of large language models: A survey, ACM Computing Surveys 57 (2025) 1–35.

[4] J. Lave, E. Wenger, Situated learning: Legitimate peripheral participation, Cambridge university press, 1991.

[5] A. Collins, J. S. Brown, S. E. Newman, Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics, in: Knowing, learning, and instruction, Routledge, 2018, pp. 453–494.

[6] C. Park, S. Han, X. Guo, A. Ozdaglar, K. Zhang, J.-K. Kim, Maporl: Multi-agent post-co-training for collaborative large language models with reinforcement learning, ArXiv abs/2502.18439 (2025). doi:10.48550/arXiv.2502.18439.

[7] K. Liu, Z. Chen, Z. Fu, W. Zhang, R. Jiang, F. Zhou, Y. Chen, Y. Wu, J. Ye, Structure-aware domain knowledge injection for large language models, 2025. URL: https://arxiv.org/abs/2407.16724. arXiv:2407.16724.

[8] J. Lu, W. Zhong, Y. Wang, Z. Guo, Q. Zhu, W. Huang, Y. Wang, F. Mi, B. Wang, Y. Wang, et al., Yoda: Teacher-student progressive learning for language models, arXiv preprint arXiv:2401.15670 (2024).

[9] J. Li, X. Dong, Y. Liu, Z. Yang, Q. Wang, X. Wang, S. Zhu, Z. Jia, Z. Zheng, Reflectevo: Improving meta introspection of small llms by learning self-reflection, arXiv preprint arXiv:2405.16475 (2024).

[10] J. Liu, Z. Huang, T. Xiao, J. Sha, J. Wu, Q. Liu, S. Wang, E. Chen, Socraticlm: Exploring socratic personalized teaching with large language models, Advances in Neural Information Processing Systems 37 (2024) 85693–85721.

[11] Y. Wan, A. Ajith, Y. Liu, K. Lu, C. Grazian, B. Hoex, W. Zhang, C. Kit, T. Xie, I. T. Foster, Sciqag: A framework for auto-generated scientific question answering dataset with fine-grained evaluation, arXiv preprint arXiv:2405.09939 (2024).
[12] Q. Zhong, L. Ding, X. Cai, J. Liu, B. Du, D. Tao, Kaft: Knowledge-aware fine-tuning for boosting llms' domain-specific question-answering performance, arXiv preprint arXiv:2405.15480 (2024).
[13] K. Kujanpää, H. Valpola, A. Ilin, Knowledge injection via prompt distillation, 2024. URL: https://arxiv.org/abs/2412.14964. arXiv:2412.14964.
[14] G. Kim, D. Jang, E. Yang, Promptkd: Distilling student-friendly knowledge for generative language models via prompt tuning, arXiv preprint arXiv:2402.12842 (2024).
[15] A. Kendapadi, K. Zaman, R. R. Menon, S. Srivastava, Interact: Enabling interactive, question-driven learning in large language models, arXiv preprint arXiv:2402.11388 (2024).
[16] Y. Tian, Y. Han, X. Chen, W. Wang, N. V. Chawla, Beyond answers: Transferring reasoning capabilities to smaller llms using multi-teacher knowledge distillation, in: Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining, 2025, pp. 251–260.
[17] Z. Li, Y. Ji, R. Meng, D. He, Learning from committee: Reasoning distillation from a mixture of teachers with peer-review, arXiv preprint arXiv:2401.03663 (2024).
[18] X. Ning, Z. Wang, S. Li, Z. Lin, P. Yao, T. Fu, M. Blaschko, G. Dai, H. Yang, Y. Wang, Can llms learn by teaching for better reasoning? a preliminary study, Advances in Neural Information Processing Systems 37 (2024) 71188–71239.
[19] D. Dinucu-Jianu, J. Macina, N. Daheim, I. Hakimi, I. Gurevych, M. Sachan, From problem-solving to teaching problem-solving: Aligning llms with pedagogy using reinforcement learning, arXiv preprint arXiv:2505.15607 (2025).
[20] P. Wang, Z. Li, N. Zhang, Z. Xu, Y. Yao, Y. Jiang, P. Xie, F. Huang, H. Chen, Wise: Rethinking the knowledge memory for lifelong model editing of large language models, Advances in Neural Information Processing Systems 37 (2024) 53764–53797.
[21] A. Albalak, D. Phung, N. Lile, R. Rafailov, K. Gandhi, L. Castricato, A. Singh, C. Blagden, V. Xiang, D. Mahan, et al., Big-math: A large-scale, high-quality math dataset for reinforcement learning in language models, arXiv preprint arXiv:2502.17387 (2025).