Code-Mixed Mathematical Reasoning in Small Language Models via Synthetic Chain-of-Thought Fine-Tuning

A Comprehensive Literature Review

Debayan Dutta

1. Introduction

The advent of Large Language Models (LLMs) has fundamentally altered the landscape of artificial intelligence, offering unprecedented capabilities in natural language understanding, generation, and reasoning. Models such as GPT-4, Claude 3.5, and Llama 3 have demonstrated performance that rivals or exceeds human capabilities on a plethora of standardized benchmarks. However, a critical examination of the prevailing literature reveals a persistent and deepening asymmetry in the distribution of these capabilities. The progress in NLP has been inextricably linked to the dominance of English, both as the primary language of pre-training corpora and as the default "lingua franca" for instruction tuning and alignment. This Anglocentric hegemony creates a significant barrier for the deployment of advanced AI systems in multilingual societies, particularly in the Global South, where communication is rarely monolingual.

This report presents an exhaustive literature review addressing a specific, high-stakes frontier in NLP: **Code-Mixed Mathematical Reasoning**. It focuses on the linguistic phenomenon of code-mixing (CM)—specifically the interleaving of Hindi and English ("Hinglish")—which serves as the dominant mode of communication for hundreds of millions of speakers in India.[1] While code-mixing is ubiquitous in informal digital discourse and educational settings, current LLMs exhibit a stark performance degradation when tasked with performing rigorous mathematical reasoning in this modality. The intersection of this linguistic challenge with the constraints of **Small Language Models (SLMs)** and the emerging methodology of **Synthetic Chain-of-Thought (CoT) Fine-Tuning** constitutes the core scope of this analysis.

The review is structured to meticulously dissect the layers of this problem. It begins by analyzing the "Translation Barrier" and the associated "Translation Tax" that English-centric models incur when processing code-mixed text. It then critiques the historical trajectory of benchmarks, highlighting the critical shift from surface-level Natural Language Understanding (NLU) tasks—such as Part-of-Speech tagging—to deep Natural Language Reasoning (NLR) tasks like mathematical problem solving. Central

to this theoretical exploration is the "Language of Thought" hypothesis, which is re-evaluated in the context of transformer-based architectures to understand whether models possess a language-agnostic conceptual space or are fundamentally tethered to English. Finally, the report synthesizes cutting-edge methodologies from 2024 and 2025, specifically examining the application of Matrix Language Frame (MLF) theory in synthetic data generation and the use of Parameter-Efficient Fine-Tuning (PEFT) techniques like QLoRA to empower SLMs.

1.1 The Sociolinguistic Reality: Code-Mixing as the Norm

Code-mixing is not an aberration or a sign of linguistic deficiency; it is a sophisticated communicative strategy employed by bilinguals to optimize information transfer, express identity, and manage social power dynamics.[2] In the Indian educational context, particularly in Science, Technology, Engineering, and Mathematics (STEM), code-mixing is the *de facto* medium of instruction. Teachers frequently alternate between English technical terms (the "Embedded Language") and Hindi syntactic structures (the "Matrix Language") to explain complex concepts.[3] Consequently, an AI tutor or reasoning assistant that functions solely in monolingual English or monolingual Hindi fails to meet the pedagogical needs of the student population. The literature identifies this mismatch as a "pedagogical gap," where the tool's linguistic rigidity inhibits the learner's cognitive fluidity.[5]

1.2 The Technological Constraint: The Case for SLMs

While massive proprietary models (e.g., GPT-4) may possess sufficient capacity to overcome some code-mixing challenges through sheer scale, their deployment in the Indian context is hindered by cost, latency, and data privacy concerns. The literature emphasizes the necessity of **Small Language Models (SLMs)**—typically in the 7 billion to 8 billion parameter range—which can be deployed locally or on affordable cloud infrastructure.[7] However, these smaller models suffer disproportionately from the "Translation Tax," lacking the surplus capacity to perform implicit translation and reasoning simultaneously. This necessitates a targeted intervention: fine-tuning these models not just to *understand* Hinglish, but to *reason* in it via explicit Chain-of-Thought methodologies.[9]

---

2. The Anglocentric Hegemony: The Translation Barrier and Translation Tax

The structural bias of contemporary LLMs is rooted in their training data. With English constituting the overwhelming majority of high-quality pre-training tokens (often >90% of reasoning data), models develop a dependency on English representations for complex cognitive tasks. This dependency manifests as a "Translation Barrier" when the model is confronted with code-mixed input, forcing it to engage in an inefficient internal conversion process.

2.1 Defining the Translation Tax

The "Translation Tax" is defined in recent literature as the aggregate loss in performance, efficiency, and

fidelity incurred when a model processes a prompt in a non-dominant language compared to a semantically equivalent prompt in a dominant language (English).[11]

- **Performance Penalty:** Research by *Saji et al. (2025)* and others indicates that reasoning accuracy drops significantly—often by 20% to 40%—when mathematical problems are presented in code-mixed formats versus standard English.[13] This drop is not merely due to vocabulary unknowns (unknown tokens) but due to the "tax" levied on the model's attention mechanism, which must simultaneously resolve syntactic ambiguity and perform logical deduction.
- **Inference Latency:** The tax is also computational. Even if the model successfully answers, it often requires more tokens to generate a coherent response in a code-mixed language or requires deeper processing layers to resolve the mixed syntax, increasing latency and energy consumption—a critical factor for SLM deployment on edge devices.[11]

2.2 The Mechanism of Semantic Drift

The most pernicious component of the Translation Barrier is "Semantic Drift." This occurs when the implicit or explicit translation of the query distorts the logical premises of the problem. Unlike simple translation errors (e.g., wrong word choice), semantic drift alters the *truth conditions* of the statement.

Table 1: Typology of Semantic Drift in Code-Mixed Mathematics

Type of Drift	Mechanism	Example (Hinglish → Model Interpretation)	Consequence
Lexical Ambiguity	A code-mixed word resembles an English word with a different meaning.	*"Mere paas **do** apples hain."* (Hindi 'do' = two) → Interpreted as English verb "do".	The numerical value "2" is lost. The model treats the sentence as a command or grammatically incorrect English, failing to extract the quantity. [15]
Syntactic Misalignment	The word order (SOV in Hindi vs SVO in English) confuses the subject-object relationship during internal translation.	*"Ram ne Sita ko ball di."* (Ram gave ball to Sita) → Interpreted as "Ram Sita ball..."	The directionality of the transaction is lost. In a math problem involving exchange (e.g., "A gives B 5 coins"), this reverses the logic. [15]

Operator Confusion	Logical connectives in the matrix language are ignored or mistranslated.	*"Price 50 se **kam** hai."* (Price is **less** than 50) $\rightarrow$ "kam" (less) is treated as a noise token or proper noun.	The inequality operator ($<$) is lost. The model solves for equality or hallucination. [17]
Negation Blindness	Negation particles in the embedded language are overlooked.	*"Box mein red ball **nahi** hai."* (There is **no** red ball in the box).	The model attends to "red ball" and ignores "nahi," assuming existence rather than absence. [14]

Recent studies highlight that this drift is exacerbated in "Reasoning" tasks compared to "Generation" tasks. In creative writing, a slight drift might be acceptable or even poetic; in mathematics, a drift from "less than" to "equal to" is a catastrophic failure.[15]

2.3 The "Lost in Translation" Effect

The paper *"The Reasoning Lingua Franca: A Double-Edged Sword for Multilingual AI"* (2025) provides rigorous empirical evidence of this phenomenon. The authors demonstrate that when LRMs (Large Reasoning Models) are forced to reason in the question's language (e.g., Hinglish), they avoid the errors introduced by translating the question into English. However, because their reasoning capabilities in Hinglish are weak, they make logical errors. Conversely, when they translate to English to reason, they suffer from "translation errors" (Semantic Drift). This catch-22 is termed the "Lost in Translation" effect.[13] The study concludes that the optimal path forward is not to improve translation, but to improve **native reasoning capabilities** in the target language, thereby removing the translation step entirely.

2.4 The Alignment Tax

A crucial and counter-intuitive finding in 2024-2025 literature is the negative impact of safety alignment on multilingual reasoning. This is termed the **Alignment Tax**.

- **Mechanism:** Post-training stages like Supervised Fine-Tuning (SFT) and Reinforcement Learning (RLHF) typically use English-heavy preference datasets. These datasets punish "unusual" syntax and reward standard, safe English prose.
- **Impact on Code-Mixing:** Code-mixed text, by definition, violates standard English grammar. An aggressively aligned model may treat code-mixed logical steps as "hallucinations" or "poor quality text," actively suppressing the generation of such tokens. *Yang et al. (2025)* show that standard SFT improves English math scores but degrades multilingual math scores, indicating that alignment consumes the model's "multilingual capacity".[12]
- **Implication:** To fix this, one cannot simply use off-the-shelf aligned models (like Llama-3-Instruct); one must perform specialized fine-tuning that re-introduces code-mixing as a valid, high-reward

modality for reasoning.[20]

---

3. The Benchmark Landscape: From NLU to NLR

To understand the current research gap, one must analyze the evolution of evaluation metrics. The history of Code-Mixed NLP is characterized by a "Metric Gap"—a discrepancy between what was measured and what was actually required for intelligent systems.

3.1 The Era of NLU: LinCE and GLUECoS

For the majority of the last decade, the field was defined by benchmarks like **LinCE** (Linguistic Code-Switching Evaluation) [21] and **GLUECoS** (General Language Understanding Evaluation for Code-Switching).[22]

- **Scope:** These benchmarks focused on **Natural Language Understanding (NLU)**. Tasks included Language Identification (LID), Part-of-Speech (POS) tagging, Sentiment Analysis, and Named Entity Recognition (NER).
- **The Limitation:** These tasks are fundamentally *local*. A model can achieve near-perfect accuracy on POS tagging by looking at a window of 3-5 words. It does not need to maintain a long-range dependency or perform symbolic manipulation.
- **The "Clever Hans" Effect:** High scores on these benchmarks created a false sense of security. Researchers assumed that because models achieved 90% F1 on Hinglish NER, they "understood" Hinglish. In reality, the models were merely performing sophisticated pattern matching. They could identify that "Apple" is a company, but they could not answer *"If Apple sells 5 phones and Samsung sells 3, how many total?"* in Hinglish.[23]
- **Obsolescence:** As of 2025, these benchmarks are considered insufficient for evaluating GenAI capabilities. They fail to test **generative coherence**, **logical consistency**, or **arithmetic fidelity**.[23]

3.2 The Shift to NLR: CodeMixBench and CM-GSM8K

The introduction of **CodeMixBench** (EMNLP 2025) and specifically the **CM-GSM8K** dataset represents a paradigm shift towards **Natural Language Reasoning (NLR)**.[25]

- **CM-GSM8K Construction:** This dataset was created by deriving 4,367 math problems from the canonical GSM8K test set. The construction utilized a rigorous "Translate-then-Verify" methodology using GPT-4, ensuring that the mathematical logic remained intact while the linguistic surface form was transformed into code-mixed text (e.g., Hindi-English, Tamil-English).[25]
- **The Reality Check:** Initial evaluations on CM-GSM8K revealed the "Reasoning Gap." Models that excelled at English GSM8K (e.g., Llama-3-70B) saw performance drops of 30-50% on the code-mixed version. The drop was even more precipitous for SLMs (Llama-3-8B), which often collapsed to near-random performance.[25]
- **Failure Modes:** The benchmarks highlighted specific failure modes unique to NLR:
    1. **Code-Switching Loop:** The model gets stuck generating a mixed sentence and repeats a phrase endlessly.

2. **Language Hallucination:** The model answers a Hindi question in French or Spanish, triggered by a token that exists in multiple languages.
3. **Logical Inconsistency:** The model sets up the equation correctly in Step 1 (in English) but misinterprets a Hindi constraint in Step 2, leading to a wrong answer.[26]

Table 2: Comparison of Code-Mixed Benchmarks

Feature	LinCE / GLUECoS	CodeMixBench / CM-GSM8K
Primary Focus	NLU (Understanding & Classification)	NLR (Reasoning & Generation)
Task Types	POS, NER, LID, Sentiment	Math Word Problems, Logic, Reasoning
Cognitive Load	Low (Local context, Pattern matching)	High (Multi-step logic, Symbolic manipulation)
Output Format	Labels / Tags	Chain-of-Thought / Free Text
Relevance to AI Tutors	Low (Backend processing only)	High (Core interaction capability)
Current SOTA Gap	Solved (High Accuracy)	Unsolved (High Error Rate in SLMs)

3.3 The Scarcity of Fine-Tuning Research

While the *evaluation* side has matured with CM-GSM8K, the *training* side remains underdeveloped. The literature reveals a stark scarcity of work focusing on **fine-tuning** SLMs specifically for code-mixed reasoning.

- **The Gap:** Most existing research relies on "Prompt Engineering" (Few-Shot Prompting) to elicit reasoning from pre-trained models. There are very few papers that propose specific **Instruction Tuning** datasets or methodologies to *train* a model to reason in Hinglish from scratch.[9]
- **Why the Scarcity?** The primary bottleneck is data. Constructing a high-quality "Reasoning Trace" (Chain-of-Thought) in Hinglish is difficult. It cannot be done by simple translation (which causes semantic drift). It requires expert human annotation (expensive) or highly sophisticated synthetic generation pipelines (complex). This report argues that addressing this scarcity via synthetic

distillation is the most promising research avenue for 2026.

---

4. The Theoretical Core: The "Language of Thought" in Multilingual Models

To solve the engineering problem of code-mixed reasoning, one must engage with the cognitive science of how LLMs "think." The **Language of Thought (LoT)** hypothesis provides the theoretical framework for understanding the limitations of current models and designing better training protocols.

4.1 Fodor's Hypothesis vs. Transformer Reality

Jerry Fodor (1975) proposed that human cognition operates in "Mentalese"—a non-linguistic, symbolic representation system. Natural languages (English, Hindi) are merely input/output interfaces for this internal processor.[28]

- **LLM Behavior:** Does an LLM have a "Mentalese"? Recent research suggests that for current transformer architectures, **English serves as the effective Language of Thought**.[30]
- **Evidence:** *Shi et al. (2023)* demonstrated that LLMs often translate non-English queries into English in their intermediate layers to process them. The activation patterns for a French math problem converge with those of its English translation in the deeper layers.[31]
- **The Implication:** If English is the LoT, then any non-English input incurs a cognitive load: the cost of translation. For code-mixed input, which switches languages mid-sentence, this load is erratic and high. The model must constantly toggle its "interface" to feed the English-based "reasoning core."

4.2 Code-Mixed CoT as a New LoT

The "Reasoning Lingua Franca" paper [13] and related work on "Multilingual Chain-of-Thought" [33] propose a radical shift: **Training the model to use the target language (or code-mix) as the medium of reasoning.**

- **The Hypothesis:** If we fine-tune an LLM on reasoning traces that are natively code-mixed, we can force the model to develop "Hinglish reasoning circuits." The model learns to manipulate the symbols of Hinglish directly (e.g., treating "2 aur 2" as a logical operation) without routing through English translation.
- **Benefits:** This eliminates the Translation Tax. It aligns the model's processing with the user's input, reducing semantic drift. *Wong et al. (2023)* describe this as "Translating from natural language to the probabilistic language of thought," arguing that the closer the CoT is to the input language, the higher the fidelity of the reasoning.[30]

4.3 Pedagogical Translanguaging Parallels

This computational hypothesis mirrors the educational theory of **Translanguaging**.[34]

- **Educational Theory:** Students in multilingual classrooms learn complex subjects (like math) better

when they are allowed to use their full linguistic repertoire. Forcing them to think solely in English imposes a cognitive burden that hampers conceptual understanding.

- **AI Parallel:** Similarly, forcing an LLM to "think" in English when the input is Hinglish hampers its performance. "Pedagogical Translanguaging" for AI implies that the AI should be trained to reason in the mixed language, just as a student is encouraged to do. This theoretical alignment between human pedagogy and AI training suggests that **Code-Mixed CoT** is not just a technical hack, but a cognitively sound approach.[34]

---

5. Methodological Frontiers: Synthesizing the Solution

Given the data scarcity, the path forward relies on **Synthetic Data Generation**. The literature from 2024-2025 highlights a specific pipeline: **Teacher-Student Distillation** guided by **Linguistic Theory** and implemented via **QLoRA**.

5.1 Linguistic Theory in Data Generation: MLF vs. ECT

Generating synthetic code-mixed text is the greatest challenge. Random insertion of words leads to "syntactic garbage" that damages the model's internal grammar. To generate high-quality training data, researchers employ linguistic theories to constrain the output of Teacher models (like GPT-4).

Table 3: Comparison of Linguistic Constraints for Synthetic Data

Theory	Principle	Suitability for LLM Generation
Equivalence Constraint Theory (ECT)	Code-switching is valid only where the grammar of both languages aligns linearly (e.g., matching parse trees).	**Low.** It is too restrictive and computationally expensive to enforce. It requires perfect parsing of both languages, which is hard for low-resource languages. Data generated via ECT often feels unnatural or stilted.[16]
Matrix Language Frame (MLF)	Assumes a hierarchical structure: One language (Matrix) provides the grammatical frame (morphosyntax), while the other (Embedded) provides	**High.** This models how humans actually speak Hinglish (Hindi grammar + English nouns/verbs). LLMs can be easily prompted to follow MLF ("Keep Hindi sentence structure but use

	content words.	English math terms"). Studies show MLF-generated data yields better downstream reasoning performance.[16]

Recommendation: The literature strongly favors **MLF-based generation**. A typical prompt would be: *"You are a Hinglish expert. Translate this math problem into Hinglish. Use Hindi as the Matrix Language (providing grammar and structure) and English as the Embedded Language (for mathematical terms and numbers). Ensure the reasoning steps follow this same structure."*.[26]

5.2 Synthetic Distillation Pipeline

The **Teacher-Student** framework is the standard for overcoming the data bottleneck.[39]

1. **Selection:** Choose a high-quality English math dataset (e.g., GSM8K).
2. **Synthesis (Teacher):** Use a Teacher Model (GPT-4) with MLF prompts to generate:
   - Code-Mixed Question (Q_{CM})
   - Code-Mixed Chain-of-Thought (CoT_{CM})
   - Final Answer (A)
3. **Filtration (The Judge):** Use a verifier model to check the validity.
   - *Answer Consistency:* Does A match the original English answer?
   - *Linguistic Fidelity:* Is the text valid Hinglish or gibberish? (Can be checked by a smaller fine-tuned BERT model or LLM-as-a-judge).
4. **Distillation (Student):** Fine-tune the SLM (Student) on the pair (Q_{CM} , CoT_{CM}) to predict the reasoning trace and answer.

5.3 Enabling SLMs: QLoRA and Unsloth

The target architecture for this research is the **Small Language Model (SLM)** (e.g., Llama-3-8B).

- **Why SLMs?** For the "Next Billion Users" in India, inference cost is the primary constraint. 70B models are too expensive to run for free educational apps. 8B models can run on consumer GPUs or even advanced mobile hardware (quantized).
- **QLoRA (Quantized Low-Rank Adaptation):** This technique freezes the base model in 4-bit precision and trains only a small set of adapter weights. This reduces memory usage by ~75%, allowing fine-tuning on a single 16GB/24GB GPU (accessible via free Colab or Kaggle).[10]
- **Unsloth:** The **Unsloth** library (released/popularized in 2024-2025) is critical. It optimizes the QLoRA backward pass, speeding up training by 2x and reducing memory further. This makes the iterative experimentation with different MLF prompts feasible for researchers with limited budgets.[41]
- **Impact:** The combination of **Synthetic CoT + QLoRA + Unsloth** is the "democratization stack" that allows researchers to build SOTA code-mixed reasoning models without access to H100

clusters.

---

6. Case Study & Implications: The Indian Context

The technical research has profound implications for the educational landscape in India.

6.1 The EdTech Reality

India is witnessing an EdTech boom, with platforms like **Physics Wallah** and **Khan Academy India** reaching millions. However, a significant gap remains.

- **The "Hinglish" Gap:** While content is often delivered in Hinglish (video lectures), the *assessments* and *AI doubtsolvers* often struggle with it. A student asking a doubt in natural Hinglish is often forced to translate it to English to get a good answer from ChatGPT or an API wrapper.[4]
- **Khan Academy's Limitation:** Literature notes that while Khan Academy has localized content (Hindi videos), the interactive AI components often lack the nuance of code-mixed reasoning. They may support pure Hindi or pure English, but fail on the fluid mix that students actually use.[44]

6.2 Democratizing STEM Education

A fine-tuned SLM that can reason in Hinglish acts as a **force multiplier**.

- **Access:** It allows students in Tier-2 and Tier-3 cities, who are often comfortable in Hinglish but intimidated by formal English, to engage with complex STEM concepts without a linguistic barrier.
- **Cognitive Offloading:** By removing the need to translate their thoughts to English, the AI allows students to focus their cognitive load on the *mathematics*, not the *language*.[3]
- **Sovereign AI:** Developing these capabilities within India, using open weights (Llama/Mistral) and local fine-tuning, reduces dependency on Western APIs and ensures data privacy for Indian students.[8]

---

7. Conclusion and Future Directions

This exhaustive review identifies **Code-Mixed Mathematical Reasoning** as a critical, under-addressed frontier in NLP. The transition from NLU-focused benchmarks to reasoning-centric ones like **CM-GSM8K** has exposed the fragility of English-centric LLMs, revealing a deep "Translation Barrier" and "Translation Tax."

7.1 Summary of Findings

1. **The Barrier is Cognitive:** The failure of LLMs in code-mixing is not just a vocabulary issue; it is a failure of the "Language of Thought." English-centric reasoning circuits cannot process mixed syntax without lossy translation.
2. **Benchmarks are Evolving:** The field has moved from "Can you identify Hindi?" (LinCE) to "Can

you solve calculus in Hinglish?" (CM-GSM8K). Current models are failing this new test.

3. **Data is the Key:** The scarcity of training data is the primary bottleneck. **Synthetic Distillation**, guided by **MLF Theory**, is the only scalable solution.
4. **SLMs are the Vehicle:** For impact in the Global South, solutions must be lightweight. **QLoRA** and **Unsloth** provide the technical means to embed high-level reasoning into small, efficient models.

7.2 Recommendations for Research

- **Construct MLF-Constrained Datasets:** Future work must focus on generating massive-scale, high-quality synthetic CoT datasets using rigorous linguistic constraints.
- **Investigate "Reasoning Transfer":** Research should quantify exactly how much "reasoning capability" can be transferred from English to Hinglish via distillation. Does learning to solve math in Hinglish improve the model's English math skills (positive transfer) or hurt them (alignment tax)?
- **Develop "Native" Evaluation Metrics:** We need metrics beyond accuracy. We need to measure "Reasoning Faithfulness"—did the model actually use the Hinglish logic, or did it translate to English and back?

The convergence of linguistic theory, synthetic data generation, and efficient fine-tuning offers a tangible path to breaking the English hegemony in AI. By teaching Small Language Models to *think* in the language of their users, we can build a generation of AI tools that are truly globally inclusive.

(End of Report)

Works cited

1. Beyond Monolingual Assumptions: A Survey of Code-Switched NLP in the Era of Large Language Models - arXiv, accessed on February 11, 2026, https://arxiv.org/html/2510.07037v1
2. A Literature Survey on AI-Driven Code-Mixed Text Analysis and Normaliz - Taylor & Francis eBooks, accessed on February 11, 2026, https://www.taylorfrancis.com/chapters/edit/10.1201/9781032724508-7/literature-survey-ai-driven-code-mixed-text-analysis-normalization-poonam-gupta-parvinder-singh
3. Building Educational Technologies for Code-Switching: Current Practices, Difficulties and Future Directions - MDPI, accessed on February 11, 2026, https://www.mdpi.com/2226-471X/7/3/220
4. HinglishEval: Evaluating the Effectiveness of Code-generation Models on Hinglish Prompts - AI @ IISc, accessed on February 11, 2026, https://kiac.iisc.ac.in/wp-content/uploads/2025/06/HinglishEval.pdf
5. (PDF) Artificial Intelligence Tutors in India's Classrooms: A Comparative Exploration of Language Education through Adaptive Systems - ResearchGate, accessed on February 11, 2026, https://www.researchgate.net/publication/398573284_Artificial_Intelligence_Tutors_in_India's_Classrooms_A_Comparative_Exploration_of_Language_Education_through_Adaptive_Systems

6. Mitigating Conceptual Learning Gaps in Mixed-Ability Classrooms: A Learning Analytics-Based Evaluation of AI-Driven Adaptive Feedback for Struggling Learners - MDPI, accessed on February 11, 2026, https://www.mdpi.com/2076-3417/15/8/4473
7. Small Language Models for Curriculum-based Guidance - arXiv, accessed on February 11, 2026, https://arxiv.org/html/2510.02347v1
8. An Open Door: AI Innovation in the Global South amid Geostrategic Competition - CSIS, accessed on February 11, 2026, https://www.csis.org/analysis/open-door-ai-innovation-global-south-amid-geostrategic-competition
9. Improving Weak-to-Strong Generalization with Reliability-Aware Alignment - ResearchGate, accessed on February 11, 2026, https://www.researchgate.net/publication/381770956_Improving_Weak-to-Strong_Generalization_with_Reliability-Aware_Alignment
10. A Practical Guide to Fine-Tuning Small Language Models - Omdena, accessed on February 11, 2026, https://www.omdena.com/blog/fine-tuning-small-language-models
11. FRANK MORALES - Boeing Associate Technical Fellow at The Boeing Company - Thinkers360, accessed on February 11, 2026, https://www.thinkers360.com/tl/profiles/view/25153
12. language imbalance driven rewarding for multilingual self-improving, accessed on February 11, 2026, https://nlpr.ia.ac.cn/cip/ZongPublications/2025/2025-YangWen-ICLR.pdf
13. The Reasoning Lingua Franca: A Double-Edged Sword for Multilingual AI - ResearchGate, accessed on February 11, 2026, https://www.researchgate.net/publication/396847749_The_Reasoning_Lingua_Franca_A_Double-Edged_Sword_for_Multilingual_AI
14. The Reasoning Lingua Franca: A Double-Edged Sword for Multilingual AI - arXiv, accessed on February 11, 2026, https://arxiv.org/html/2510.20647v3
15. Beyond Monolingual Assumptions: A Survey on Code-Switched NLP in the Era of Large Language Models across Modalities - arXiv, accessed on February 11, 2026, https://arxiv.org/html/2510.07037v5
16. Lost in the Mix: Evaluating LLM Understanding of Code-Switched Text - arXiv, accessed on February 11, 2026, https://arxiv.org/html/2506.14012v1
17. Multilingual Reasoning Traces - Emergent Mind, accessed on February 11, 2026, https://www.emergentmind.com/topics/multilingual-reasoning-traces
18. LEARN GLOBALLY, SPEAK LOCALLY: BRIDGING THE GAPS IN MULTILINGUAL REASONING - OpenReview, accessed on February 11, 2026, https://openreview.net/pdf/8d718ac3c524967a3aeab7adaf11c89c4725e235.pdf
19. Learn Globally, Speak Locally: Bridging the Gaps in Multilingual Reasoning - arXiv, accessed on February 11, 2026, https://arxiv.org/html/2507.05418v1
20. LANGUAGE IMBALANCE DRIVEN REWARDING FOR MULTILINGUAL SELF-IMPROVING - OpenReview, accessed on February 11, 2026, https://openreview.net/pdf?id=Kak2ZH5Itp
21. GLUECoS: An Evaluation Benchmark for Code-Switched NLP - ACL ..., accessed on February 11, 2026, https://aclanthology.org/2020.acl-main.329/
22. GLUECoS: An Evaluation Benchmark for Code-Switched NLP | Request PDF, accessed

on February 11, 2026, https://www.researchgate.net/publication/343300735_GLUECoS_An_Evaluation_Benchmark_for_Code-Switched_NLP
23. Limitations of large language models in clinical problem-solving arising from inflexible reasoning - PMC, accessed on February 11, 2026, https://pmc.ncbi.nlm.nih.gov/articles/PMC12606185/
24. Line Goes Up? Inherent Limitations of Benchmarks for Evaluating Large Language Models, accessed on February 11, 2026, https://arxiv.org/html/2502.14318v1
25. CodeMixBench: Evaluating Code-Mixing Capabilities of LLMs Across 18 Languages - arXiv, accessed on February 11, 2026, https://arxiv.org/html/2507.18791v2
26. CodeMixBench: Evaluating Code-Mixing Capabilities of LLMs Across 18 Languages - ACL Anthology, accessed on February 11, 2026, https://aclanthology.org/2025.emnlp-main.109.pdf
27. On Code-Induced Reasoning in LLMs - OpenReview, accessed on February 11, 2026, https://openreview.net/forum?id=LIv0bfJZIi
28. Origins of numbers: a shared language-of-thought for arithmetic and geometry? - PMC - NIH, accessed on February 11, 2026, https://pmc.ncbi.nlm.nih.gov/articles/PMC7618345/
29. Explainability Through Systematicity: The Hard Systematicity Challenge for Artificial Intelligence - PMC, accessed on February 11, 2026, https://pmc.ncbi.nlm.nih.gov/articles/PMC12307450/
30. On the Fundamental Limits of LLMs at Scale - arXiv, accessed on February 11, 2026, https://arxiv.org/html/2511.12869v1
31. Embers of autoregression show how large language models are shaped by the problem they are trained to solve | PNAS, accessed on February 11, 2026, https://www.pnas.org/doi/10.1073/pnas.2322420121
32. Cross-lingual Prompting: Improving Zero-shot Chain-of-Thought Reasoning across Languages | Request PDF - ResearchGate, accessed on February 11, 2026, https://www.researchgate.net/publication/376401517_Cross-lingual_Prompting_Improving_Zero-shot_Chain-of-Thought_Reasoning_across_Languages
33. The Reasoning Lingua Franca: A Double-Edged Sword for Multilingual AI - arXiv, accessed on February 11, 2026, https://arxiv.org/html/2510.20647v2
34. (PDF) AI-Driven Translanguaging: Enhancing Plurilingual Proficiency in EFL Classrooms, accessed on February 11, 2026, https://www.researchgate.net/publication/398262047_AI-Driven_Translanguaging_Enhancing_Plurilingual_Proficiency_in_EFL_Classrooms
35. «APPLIED LINGUISTICS-3D: LANGUAGE, IT, ELT», accessed on February 11, 2026, https://conf.ztu.edu.ua/wp-content/uploads/2025/06/povnyj-tekst.pdf
36. Designing an AI-Supported Framework for Literary Text Adaptation in Primary Classrooms, accessed on February 11, 2026, https://www.mdpi.com/2673-2688/6/7/150
37. Code-Switching Curriculum Learning for Multilingual Transfer in LLMs - ACL Anthology, accessed on February 11, 2026, https://aclanthology.org/2025.findings-acl.407.pdf
38. Leveraging Large Language Models for Code-Mixed Data Augmentation in Sentiment Analysis - arXiv, accessed on February 11, 2026, https://arxiv.org/html/2411.00691v1
39. Daily Papers - Hugging Face, accessed on February 11, 2026,

https://huggingface.co/papers?q=latent%20semantic%20analysis
40. Synthetic data distillation enables the extraction of clinical information at scale - PMC - NIH, accessed on February 11, 2026, https://pmc.ncbi.nlm.nih.gov/articles/PMC12065832/
41. QLoRA Fine-Tuning with Unsloth: A Complete Guide - Medium, accessed on February 11, 2026, https://medium.com/@matteo28/qlora-fine-tuning-with-unsloth-a-complete-guide-8652c9c7edb3
42. Fine-tune Llama 3.1 Ultra-Efficiently with Unsloth - Hugging Face, accessed on February 11, 2026, https://huggingface.co/blog/mlabonne/sft-llama3
43. Fine-tuning LLMs Guide | Unsloth Documentation, accessed on February 11, 2026, https://unsloth.ai/docs/get-started/fine-tuning-llms-guide
44. LANGUAGE IN INDIA, accessed on February 11, 2026, https://www.languageinindia.com/nov2025/v25i11nov2025.pdf