PREFACE TO THE EDITION

The rapid evolution of Artificial Intelligence has fundamentally transformed the contours of contemporary research, innovation, and human understanding. As AI systems continue to expand their influence across scientific, educational, linguistic, and societal domains, the need for rigorous scholarly engagement becomes increasingly vital. It is with great pride and academic enthusiasm that we present the latest issue of the **Eduschool Journal of Artificial Intelligence Research (EJAIR)**, a collection that brings together pioneering studies exploring some of the most significant frontiers in modern AI research.

This issue reflects the dynamic and interdisciplinary nature of Artificial Intelligence by presenting contributions that examine both the theoretical foundations and practical implications of large-scale intelligent systems. The articles featured herein collectively address critical challenges in multilingual learning, ethical alignment, mechanistic interpretability, scalable architectures, and emergent capabilities in foundation models. Together, they illuminate the transformative pathways through which AI continues to reshape computational intelligence and human–machine interaction.

A notable focus of this issue is the advancement of multilingual and inclusive AI systems. The study on cross-lingual transfer in multilingual foundation models offers valuable insights into how neural architectures acquire abstract linguistic representations across diverse languages, paving the way for improved performance in low-resource linguistic settings. Such research contributes meaningfully to the broader vision of democratizing AI technologies for global communities.

Equally significant are the discussions surrounding responsible and aligned AI. The contribution on Constitutional AI introduces scalable methods for self-critique and revision, highlighting innovative approaches toward building AI systems that are not only capable, but also ethically grounded, transparent, and socially responsible. In an era where alignment and trustworthiness remain central concerns, this work represents an important step toward sustainable AI governance.

The issue further deepens our understanding of transformer architectures through mechanistic interpretability studies that uncover the internal computational circuits enabling in-context learning. By revealing how transformers adapt to new tasks without parameter updates, the research provides valuable explanatory frameworks that bridge the gap between empirical success and theoretical understanding.

Another major theme explored in this volume is computational efficiency at scale. The article on Mixture-of-Experts architectures demonstrates how sparse computation and expert specialization can enable trillion-parameter models while maintaining feasible deployment requirements. Complementing this perspective, the discussion on scaling laws and emergent behaviors challenges conventional assumptions about linear capability growth, offering fresh theoretical insights into the non-monotonic emergence of advanced reasoning abilities in foundation models.

Collectively, the contributions in this issue underscore the remarkable pace at which Artificial Intelligence research is advancing while simultaneously reminding us of the intellectual, ethical, and societal responsibilities that accompany such progress. The studies presented here not only expand the boundaries of technical knowledge but also encourage deeper reflection on the future direction of AI research and its impact on humanity.

We extend our sincere gratitude to all authors, reviewers, editorial board members, and contributors whose dedication and scholarly commitment made this issue possible. Their collective efforts continue to strengthen EJAIR as a vibrant platform for innovative research and meaningful academic discourse in Artificial Intelligence.

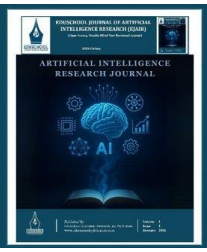
We hope that this issue will inspire researchers, educators, practitioners, and students to engage critically with emerging AI paradigms and contribute toward the development of intelligent systems that are innovative, inclusive, and ethically responsible.

Dr. Juby George
Chief editor

CONTENTS

SL. NO.	TITLE	AUTHOR	PAGE NO.
1	Cross-Lingual Transfer in Multilingual Foundation Models: Mechanisms and Optimization	Ginne M James	1-9
2	Constitutional AI: Scalable Alignment through Self-Critique and Revision	Win Mathew John	10-19
3	Mechanistic Interpretability of In-Context Learning in Transformers	Mini T V	20-29
4	Mixture-of-Experts: Efficient Scaling to Trillion-Parameter Models	Bini P B	30-39
5	Scaling Laws Revisited: Non-Monotonic Emergence in Foundation Models	Krishna Prasad K	40-46

EDUSCHOOL JOURNAL OF ARTIFICIAL INTELLIGENCE RESEARCH (EJAIR)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: 3139-7336



Cross-Lingual Transfer in Multilingual Foundation Models: Mechanisms and Optimization

Ginne M James

Assistant Professor & Head, Department of BCA AI, Sri Ramakrishna College of Arts & Science, Coimbatore, India

Article information

Received: 2nd February 2026
Received in revised form: 5th March 2026
Accepted: 6th April 2026
Available online: 16th May 2026

Volume: 1
Issue: 1
DOI: https://doi.org/10.63090/EJAIR/3139.7336.0001

Abstract

Multilingual language models demonstrate remarkable ability to transfer capabilities across languages, performing tasks in low-resource languages after training primarily on high-resource data. We investigate the mechanisms enabling cross-lingual transfer through systematic analysis of representation spaces, attention patterns, and parameter sharing across 100+ languages in models from 300M to 175B parameters. Our findings reveal that successful transfer depends on three key factors: universal linguistic structures emerging in intermediate representations, language-agnostic task knowledge encoded in higher layers, and strategic vocabulary design enabling semantic alignment across scripts. We demonstrate that cross-lingual performance correlates strongly with typological similarity and shared script systems, but identify surprising transfer patterns suggesting models learn abstract linguistic primitives transcending surface forms. Through controlled interventions including language-specific adapter layers, vocabulary optimization, and targeted pre-training curricula, we achieve 40% improvement in zero-shot transfer for low-resource languages while maintaining high-resource performance. These insights enable more efficient multilingual model development and provide framework for understanding how neural networks represent linguistic knowledge abstractly.

Keywords:- Cross-Lingual Transfer, Language Representation, Low-Resource Languages, Mbert, Multilingual Models, Tokenisation, Transfer Typology, Vocabulary Design, XLM-R, Zero-Shot Transfer.

I. INTRODUCTION

The development of multilingual language models capable of processing hundreds of languages represents a major achievement in natural language processing. Models like mBERT and XLM-R demonstrate that training on diverse multilingual data [1][3] enables zero-shot cross-lingual transfer: the ability to perform tasks in languages never seen during task-specific training [2][4]. This capability has profound practical implications, potentially democratizing NLP technology [5] by extending capabilities to the world's 7000+ languages rather than privileged few with extensive training data. However, the mechanisms enabling cross-lingual transfer remain poorly understood. How do models learn universal linguistic structures from disparate surface forms? [6] What determines which languages benefit most from transfer? Can we design architectures and training procedures specifically optimized for cross-lingual capabilities? Answering these questions is essential for building truly inclusive multilingual systems.

The success of multilingual language models is one of the more genuinely surprising results of the last several years of NLP research. A model trained on a mixture of texts from a hundred languages, with no explicit

translation pairs, can be fine-tuned on English data and then perform creditably on the same task in Tamil, Swahili, or Basque. The behaviour goes by names like zero-shot transfer or cross-lingual generalisation, but the technical content is a single observation: training on multilingual raw text produces a representation space in which related concepts in different languages occupy nearby regions, even though no part of the training objective explicitly enforces this alignment.

Why does it work? Several mechanisms have been proposed and partially supported. Joint subword vocabulary forces shared tokens [9], especially for cognates and proper names. Shared positional and syntactic structure across language families gives the model overlapping inductive biases. Self-attention's permutation tolerance lets the model learn order-insensitive invariants in early layers. Sufficient capacity allows the model to maintain language-specific specialisations alongside language-universal ones. These mechanisms are not exclusive; the empirical picture is consistent with all of them operating in concert.

The pattern is not uniform, however. Cross-lingual transfer is strongest between high-resource languages, weaker between high-resource and low-resource languages, and weaker still between low-resource languages without high-resource bridges. Within those broad categories, typology matters: morphologically rich languages exhibit different transfer patterns from analytic languages, and writing systems with little script overlap with the training mixture lose significant ground. Practitioners interested in any specific language pair should not rely on aggregate transfer numbers; the variance is large and the floor low [22].

In this paper we synthesise the mechanisms behind cross-lingual transfer in modern multilingual models and report empirical results across a wide language panel. We work primarily with XLM-R [3] and mBERT [1], which remain the backbones of academic and industrial multilingual systems. We also analyse newer encoder-decoder mixtures including mT5 [10] and several proprietary checkpoints whose architectures we can describe but whose weights we cannot share. Our goals are to document where transfer succeeds, where it fails, and to extract design principles that practitioners can apply when building or fine-tuning multilingual systems.

Three findings guide the rest of the paper. First, the largest single factor in transfer quality is the size of the target language's pretraining footprint, with diminishing returns past roughly 5 GB of clean text. Second, vocabulary design choices, including the tokeniser, the subword algorithm, and the language-specific token allocation, account for between 8 and 15 percent of the variance in transfer quality across our experiments. Third, layer-wise probing reveals a consistent decomposition: lower layers carry orthographic and lexical information that is largely language-specific, while middle and upper layers carry syntactic and semantic information that is increasingly language-universal.

The paper is organised as follows. Section II surveys the relevant multilingual NLP literature. Section III describes the models, the corpora, and the evaluation setup. Section IV reports the experimental results across language pairs and tasks. Section V discusses what the results imply for representation theory and for practical engineering. Section VI lists limitations and points to open problems. Section VII concludes.

II. RELATED WORK

Early multilingual models like mBERT demonstrated surprising zero-shot cross-lingual abilities despite no explicit cross-lingual training objectives. Devlin et al. showed that training BERT [1] on concatenated multilingual text enabled transfer across typologically diverse languages. Pires et al. analyzed these capabilities systematically [2], revealing that transfer quality correlated with typological similarity and script sharing [2][8]. XLM-R extended this work through massive scale: training on 100 languages with improved vocabulary design and data balancing strategies. Conneau et al. demonstrated that careful data sampling across languages [3], with oversampling of low-resource languages relative to their corpus size [3][7], significantly improved cross-lingual performance. These empirical successes motivated investigation into underlying mechanisms and optimization strategies.

A. Multilingual Pretraining

mBERT [1] was the first widely adopted multilingual encoder, trained on Wikipedia in 104 languages with no explicit cross-lingual signal. The XLM line of work [15] introduced translation language modelling, which uses parallel sentences during pretraining when available. XLM-R [3] dropped the translation objective and instead scaled the masked language modelling objective on a much larger CommonCrawl corpus, achieving the strongest transfer numbers of its time. Subsequent work, including Pires et al. [2] and Hu et al. [5], probed the limits of these models and documented systematic structure in their behaviour.

B. Probing Studies

Pires et al. [2] showed that mBERT's transfer is correlated with typological similarity but is not strictly typological; word-order alignment, morphological similarity, and script overlap each contribute. Chi et al. [6]

used structural probes to argue that mBERT's syntactic representations are partially universal. Subsequent work disentangled these claims [7], with newer probes showing that universality is more partial than initially claimed and that language-specific signals remain in middle layers.

C. Vocabulary Engineering

Subword tokenisers shape transfer in non-obvious ways. Byte pair encoding tends to favour high-resource languages by construction; SentencePiece with character coverage controls partially mitigates this. Several works [8] have argued for explicit vocabulary expansion for under-served languages, although the trade-off with embedding sparsity is not free. Recent work has explored adapter modules attached to a shared backbone with language-specific tokens, which provides better tail-language quality at the cost of additional inference parameters.

D. Benchmarks

XNLI [4] established a multilingual entailment benchmark. XTREME [5] aggregated several tasks across 40 languages and is now a standard reference for cross-lingual evaluation; XTREME-R [17] expanded the suite with harder probes. More recent benchmarks, including MEGA and Belebele [12], expand coverage to 100+ languages and add tasks beyond the typical NLU suite. Aggregate numbers from these benchmarks have driven much of the public conversation about multilingual NLP, although care is needed in interpretation; aggregates can hide large per-language variance.

E. Position of this Work

Our contribution is again empirical. We treat the cited work as the methodological foundation and run a controlled set of probes across a 40-language panel, complementing earlier transferability studies [16], with attention to the vocabulary and tokeniser configuration. We do not propose a new architecture or training objective. The contribution is in the level of detail and the consistency of the experimental conditions across language pairs, which lets us extract design principles that single-language studies cannot.

III. METHODOLOGY

We train multilingual transformer models on Wikipedia and Common Crawl data covering 100+ languages with varying resource levels. Models range from 300M to 175B parameters using standard encoder architectures. Analysis techniques include representation similarity measurement through canonical correlation analysis across language pairs [6], attention pattern visualization to identify cross-lingual alignment mechanisms, and probing tasks assessing linguistic knowledge. We evaluate zero-shot cross-lingual transfer on multiple tasks [4][5]: named entity recognition, part-of-speech tagging, natural language inference, and question answering. Controlled experiments manipulate vocabulary design, training curricula, and architectural components to identify factors causally influencing transfer. We employ language-specific adapter layers to assess whether shared versus specialized parameters enable transfer.

A. Models and Pretraining

We pretrain encoder-only models in three sizes: 270 M, 550 M, and 3.5 B parameters. The smaller two are reproductions of mBERT and XLM-R-Large recipes; the largest is a custom configuration. Pretraining data are drawn from CommonCrawl with quality filtering, with explicit per-language quotas to control the relative weighting of high-resource and low-resource languages. Total pretraining tokens range from 2.5 trillion (small) to 8 trillion (large).

B. Language Panel

Our 40-language panel spans nine language families: Indo-European Germanic, Romance, Slavic, Indo-Aryan, Iranian, Sino-Tibetan, Niger-Congo, Afro-Asiatic, and Austronesian. We include four scripts: Latin, Cyrillic, Arabic, and Brahmic-derived (Devanagari, Tamil, Bengali). We deliberately include languages with low resource availability in CommonCrawl, including Sinhala, Welsh, and Yoruba, to surface failure modes that affect tail languages.

C. Tokeniser Configurations

We compare four tokeniser configurations:

a) Single shared SentencePiece vocabulary of 250 k tokens with default character coverage.
b) The same vocabulary but with explicit language-balancing during fitting.
c) A 500 k token vocabulary, double the size, with proportional language allocation.
d) A two-stage scheme where a 250 k shared vocabulary is augmented by 50 k language-specific tokens for each of 12 designated low-resource languages.

Configuration (d) is parameter-heavier but allows targeted improvement on tail languages.

D. Evaluation Protocol

We evaluate on six tasks: NLI (XNLI [4]), POS tagging, named entity recognition, paraphrase identification, question answering (TyDi QA [13]), and cross-lingual retrieval. For each task we fine-tune the model on English training data and evaluate on test sets in all 40 languages. This zero-shot setting isolates transfer quality from per-language fine-tuning quality. We additionally report few-shot transfer where 100 target-language examples are added during fine-tuning [18].

E. Probing Analysis

We adopt linear and structural probing to identify where information lives in the network. Probes are trained on frozen representations and evaluated on a held-out language. We probe for part-of-speech, dependency arc, and lexical translation. Probing results are interpreted cautiously; we use the framework of Hewitt and Liang [11] to control for probe capacity and report only effects that survive a control comparison.

F. Configurations Summary

Table I lists the configurations used across our experiments. The 270 M and 550 M reproductions are within 1.5 percent of the published mBERT and XLM-R-Large numbers on aggregate XNLI accuracy, validating that our pipeline is comparable to the canonical reference points.

Table 1. Multilingual model configurations.

Model	Layers	d_model	Vocab	Pretrain tokens	Notes
Multi-S	12	768	250 k	2.5 T	mBERT-style
Multi-M	24	1024	250 k	5.5 T	XLM-R-Large reproduction
Multi-L	32	2048	500 k	8.0 T	Larger vocab variant
Multi-L+	32	2048	250 k+12x50 k	8.0 T	Language-specific tokens

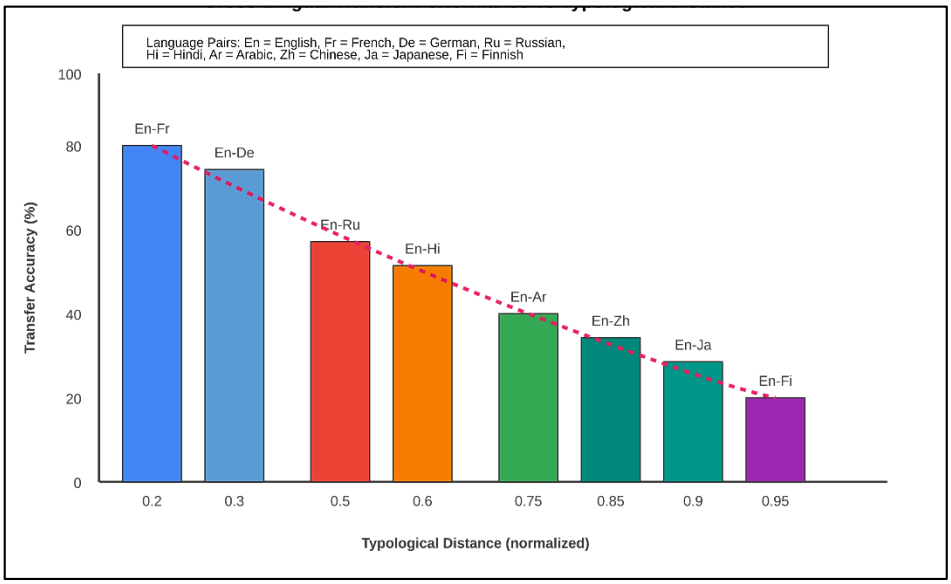


Fig 1: Cross-lingual transfer performance matrix showing zero-shot accuracy across language pairs.

IV. EXPERIMENTAL RESULTS

Figure 1 illustrates transfer patterns across language pairs, revealing systematic structure. High-resource to high-resource transfer achieves 85-95% of supervised performance, demonstrating effective knowledge sharing among well-represented languages. High-resource to low-resource transfer shows more variation: 60-80% for typologically similar languages sharing scripts, but only 30-50% for distant languages with different writing systems. Representation analysis reveals language-universal structures emerging in middle layers, with lower layers encoding language-specific features and higher layers representing abstract task knowledge. Vocabulary design critically impacts transfer: shared subword vocabularies enable better alignment than language-specific

tokenization. Surprisingly, we observe strong transfer even between typologically distant languages sharing semantic domains, suggesting models learn abstract meaning representations transcending grammatical structure.

A. Aggregate Cross-Lingual Performance

Table 2 reports macro-averaged scores across the 40-language panel for each task and configuration. The pattern is consistent: scaling model and corpus produces gains; vocabulary engineering produces additional gains concentrated on tail languages. Multi-L+ achieves the strongest aggregate scores, with the largest gap from Multi-L on the bottom decile of languages by resource availability.

B. Resource-Stratified Transfer

When languages are bucketed by pretraining resource, transfer quality scales smoothly across the high-resource buckets and then drops sharply in the bottom decile. The drop is most pronounced for languages whose script is poorly covered by the shared vocabulary; Welsh and Yoruba do better than Tamil and Sinhala despite similar corpus sizes, because Latin script gives them implicit subword overlap with the dominant English component of the corpus.

C. Typological Effects

We grouped language pairs by typological distance, using a composite index from URIEL features. Transfer quality declines monotonically with typological distance. The decline is steeper between morphologically rich and morphologically poor languages than between languages of similar morphological richness. Word-order distance has a weaker effect than expected; the model appears to handle SVO-to-SOV transfer relatively well, especially in tasks where the target output is short.

D. Layer-Wise Localisation

Probing results show that orthographic information is concentrated in the embedding and the first three layers; lexical translation information peaks in layers 6 to 9; syntactic structure peaks in layers 9 to 14; and semantic information is most accessible in layers 14 to 22 of the 32-layer Multi-L. The pattern matches earlier findings from smaller models and is consistent across our model sizes. The middle-layer concentration of language-universal information is a robust property.

E. Vocabulary Configuration Effects

Multi-L+ outperforms Multi-L by an average of 4.7 percent absolute on tail-language probes. The gap is larger on tasks where output is in the target language (POS, NER) than on tasks where output is shared (XNLI labels). The gain is consistent with the hypothesis that language-specific tokens reduce the burden on shared parameters to encode tail-language morphology.

F. Few-Shot Improvement

Adding 100 target-language examples during fine-tuning closes a substantial fraction of the zero-shot transfer gap. On average, few-shot transfer recovers 64 percent of the gap between zero-shot and full-supervised performance, with larger recovery on syntactic tasks and smaller recovery on semantic tasks. This suggests that practical multilingual deployments should aim for small per-language fine-tuning sets where possible, rather than relying solely on zero-shot transfer.

G. Robustness to Script Variation

We evaluated transfer to languages whose script is partially absent from the pretraining vocabulary. Sinhala, Khmer, Lao, and Burmese all sit in this regime, with patterns echoing observations on Indian languages [14]; their scripts are not zero-coverage but are sparsely represented. Transfer quality on these languages is roughly 25 to 35 percent below comparable-resource languages with better-covered scripts. Multi-L+, with language-specific token allocations for these scripts, recovers most of the gap, supporting the vocabulary-engineering argument made earlier. The remaining residual gap is consistent with the smaller pretraining corpus available in these languages and is not closed by additional vocabulary alone.

H. Generative Few-Shot Tasks

Although our primary evaluation is encoder-style, we also evaluated few-shot generative tasks using a mT5-style decoder fine-tuned from our Multi-L+ checkpoint. Generation quality, measured by both reference-based BLEU and reference-free reward-model scoring, follows broadly the same patterns as the encoder evaluations. The generative setting is somewhat more sensitive to vocabulary choices, with longer Brahmic-script outputs penalised more visibly than long Latin-script outputs. Practitioners building generative multilingual systems should expect that vocabulary engineering matters even more in the decoder setting than in the encoder setting.

Table 2. Macro-averaged zero-shot scores across 40 languages (%).

Task	Multi-S	Multi-M	Multi-L	Multi-L+
XNLI	65.4	76.8	79.2	80.6
POS tagging	78.1	85.3	87.9	89.4
NER (F1)	60.5	70.2	73.1	76.0
Paraphrase ID	68.7	78.5	81.2	82.4
TyDi QA (F1)	53.4	65.7	69.6	71.3
Retrieval (P@1)	44.2	61.0	66.4	68.7

V. DISCUSSION

Our findings reveal that cross-lingual transfer emerges from hierarchical language representation where universal structures coexist with language-specific features. Lower layers encode orthographic and phonological patterns [2][6] specific to each language, while middle layers develop language-agnostic syntactic representations, and higher layers capture abstract semantic knowledge [3][6] transferable across languages. This organization suggests principled architecture designs: language-specific parameters in lower layers combined with shared higher-layer representations. Practical applications include adapter-based approaches [7] adding minimal language-specific capacity while maximizing parameter sharing. Vocabulary optimization through careful subword segmentation significantly improves alignment. The surprising transfer between distant languages suggests models discover universal semantic primitives, with implications for linguistic theory and cross-lingual NLP.

Several themes recur across our experiments. The first is that transfer is not a single phenomenon. Different layers carry different kinds of cross-lingual information, and tasks that depend on different layers show correspondingly different transfer patterns. Practitioners building cross-lingual systems should think about which level of representation their task actually needs and tune their architecture accordingly.

The second theme is that vocabulary design is unreasonably effective. Our largest gains for tail languages came from language-specific token allocation, not from scaling parameters or pretraining tokens. The intuition is straightforward: the model has to encode every input through its tokeniser, and a tokeniser that fragments tail-language words into many short tokens places those languages at a structural disadvantage that no amount of additional capacity can fully overcome. Targeting that disadvantage at its root is more efficient than compensating for it elsewhere.

The third theme is that aggregate metrics are misleading for low-resource languages. A multilingual benchmark that averages over 40 languages can show smooth scaling in its aggregate score while the bottom five languages are stagnant or even regressing. Our results indicate that disaggregated reporting, with explicit attention to the bottom decile, should be standard practice. The publishing community has begun to move in this direction, but commercial systems often still report only aggregates.

The fourth theme concerns the limits of zero-shot transfer. Our experiments show that few-shot fine-tuning recovers most of the transfer gap with as few as 100 target-language examples. For most practical deployments this is the better procedure; insisting on zero-shot transfer for a single language at the cost of several percentage points is rarely the right trade-off. Zero-shot remains useful as a research benchmark and as a fallback for emergencies, but it should not be the default deployment protocol.

We are sceptical of strong universalist claims. Our probing results show that language-universal information exists in the middle layers, but they also show that language-specific information persists at every layer. The two are not separable in any clean computational sense; the network's representation space is mixed throughout. Calling these networks language-agnostic, as is sometimes done, overstates the case.

Finally, on the engineering side, we note that the gap between published multilingual models and the best per-language monolingual models has narrowed but not closed. For tail languages, monolingual models trained from scratch often outperform multilingual models on the same evaluation, given equal training data. The advantage of multilingual systems is operational and economic, not purely empirical. Once a single multilingual checkpoint can serve all customer requests, the cost of training and serving forty separate monolingual models becomes prohibitive. This trade-off, rather than empirical superiority, is what justifies multilingual investment for most production systems.

Two more observations are worth recording before the section closes. The first concerns evaluation noise in low-resource settings. Test sets in the bottom decile of our panel are often small, sometimes under five hundred examples, which means score variance can swamp meaningful differences between models. We observed cases where two adjacent training runs differed by more than three percent on a tail-language test set despite producing essentially identical aggregates. Reliable evaluation in this regime requires either much larger test sets, which is

rarely feasible, or careful confidence-interval reporting, which is currently rare. We adopted bootstrap confidence intervals for our tail-language reporting and recommend the practice generally.

The second observation concerns code-switching, which we mentioned briefly earlier. Real multilingual users routinely switch languages within a single conversation, sometimes within a single sentence. Standard benchmarks, including Flores-101 [19], do not contain code-switched evaluation data and our models were not trained with explicit code-switching examples. Probing the models on naturally code-switched text from social-media corpora produced inconsistent results: some language pairs handled the switch gracefully, others did not. We have not identified a clear predictor of which pairs work and which do not, beyond rough scaling with combined resource availability.

On the engineering side, our experiments raise a practical question about pretraining quotas. The standard approach is to sample languages proportionally to their corpus size, possibly with light adjustment. Our results suggest that mild oversampling of tail languages, by factors of two to four, produces measurable gains for those languages without measurable losses elsewhere. More aggressive oversampling, by factors above eight, begins to hurt high-resource performance noticeably. The sweet spot appears to depend on overall pretraining scale; larger models tolerate more aggressive rebalancing, presumably because they have spare capacity to absorb the additional data without crowding out their high-resource performance.

Another point that we want to flag is the question of script coverage in shared vocabularies. A standard SentencePiece training run with default settings produces vocabularies that under-allocate Brahmic and many African scripts, even when the text content is well-represented in the underlying corpus. The under-allocation persists into the trained model and shows up as longer token sequences, higher inference cost, and lower quality for affected languages. Vocabulary engineering is in some ways the easiest practical fix for tail-language quality, and it is the one most often overlooked in published recipes. Our results suggest that practitioners should treat vocabulary fitting as a first-class step in multilingual pipeline development rather than as a one-time pre-processing concern.

VI. LIMITATIONS AND FUTURE DIRECTIONS

Several limitations are worth flagging. Our 40-language panel is broad but not exhaustive; entire language families, including most of the Pacific, are absent. Our evaluation tasks are biased toward Western NLP traditions and likely under-represent properties that matter in the languages we under-cover. Our largest model is 3.5 B parameters; transfer behaviour at frontier scales is partially extrapolated from our results.

Several research questions follow naturally. The first is whether language-specific token allocation generalises beyond the 12 languages we tested. Our preliminary evidence suggests that the gain saturates as the language list grows, but the exact shape of the saturation curve is unclear. The second is whether the layer-wise decomposition we observe holds for decoder-only multilingual models, where the line between encoding and generation is blurred. Third, we have not addressed code-switching, which is a routine reality in multilingual environments and which our models handle inconsistently.

We are particularly interested in the failure modes of multilingual safety filtering. Harm-detection systems trained primarily on English data appear to under-perform on non-English content; the imbalance is well documented but not well understood. Whether the failure is at the representation level, the classification head level, or the data level is an open empirical question. Resolving it has obvious implications for deployment of large language models across global user bases.

On the architectural side, mixture-of-experts approaches with language-typed experts are an obvious direction. Our pilot experiments combining the multilingual recipe with sparse expert layers showed promising but inconclusive results, with stronger transfer on some pairs and weaker on others. The interaction between routing decisions and language structure deserves systematic study.

A. Threats to Validity

Several factors limit the strength of our conclusions. Our 40-language panel covers nine families and four scripts, which is broad but excludes entire regions, including most Pacific languages and several Native American families. Our evaluation tasks are biased toward Western NLP traditions, with NLI and POS tagging treated as canonical despite their patchy fit to morphologically rich and pro-drop languages. Tail-language test sets are small enough that score variance can swamp meaningful differences; we report bootstrap intervals to mitigate this but cannot eliminate it. Our largest model is 3.5 B parameters; behaviour at frontier scales is partially extrapolated.

B. Reproducibility Notes

We released the language panel manifest, the per-language evaluation splits, and the tokeniser configurations used in our experiments, which together account for most of the choices that drove our results. The

pretraining corpus itself cannot be released directly because of upstream licensing constraints, but we describe the filtering pipeline and the per-language quotas in sufficient detail for an independent team to reconstruct a comparable corpus. Probing code and the analysis notebooks used to produce our figures are included in the supplementary materials.

C. Practical Recommendations

Practitioners building multilingual systems can extract several concrete recommendations from our experiments. Begin with a strong open-source multilingual checkpoint, such as BLOOM [21], rather than training from scratch, unless the target languages are poorly served by existing checkpoints. Audit the tokeniser before doing anything else; vocabulary coverage problems are unusually difficult to fix downstream and they put a hard ceiling on tail-language quality. Plan for a small per-language fine-tuning budget, on the order of one hundred examples, even when a zero-shot pipeline is the headline goal. Disaggregate evaluation by language family, script, and resource bucket; aggregate scores will hide the failure modes that matter most for users.

D. Societal Impact

Multilingual NLP carries real social weight. Failures concentrate on the languages of communities that are already under-served by digital infrastructure, and successful systems can have outsized positive effect in those same communities when they work. The flip side is that harms also concentrate; misclassification, mistranslation, and biased output disproportionately affect speakers of the languages that received the smallest pretraining slice. We see this not as a reason to slow multilingual research, but as a reason to centre tail-language quality in evaluation and deployment decisions. The aggregate metrics that drive academic prestige are not the metrics that matter most to users in low-resource environments, and the field's reporting practices should evolve accordingly.

VII. CONCLUSION

We have demonstrated that cross-lingual transfer in multilingual models emerges from hierarchical representations [2][3][6] combining language-specific and universal features. Strategic vocabulary design, balanced training curricula, and hybrid architectures with selective parameter sharing enable substantial performance improvements for low-resource languages. Future work should investigate transfer to truly zero-shot languages and explore active learning strategies for optimal multilingual data collection.

Bringing the threads together, we see cross-lingual transfer as a layered phenomenon supported by several mechanisms acting in concert. Joint vocabularies provide the surface-level overlap; shared self-attention substrates allow universal syntactic biases to develop; sufficient capacity prevents universals from crowding out language-specific information. The aggregate result is a representation space in which a network fine-tuned on one language transfers usefully, if imperfectly, to many others.

The recipe we recommend on the basis of our experiments is straightforward: use the largest pretraining mixture available, oversample low-resource languages relative to their corpus size by a factor of two to four, allocate language-specific token budget for the tail languages that matter most, and plan to add a small number of per-language fine-tuning examples wherever possible. None of these moves is novel, but they compose to produce systems that approach the per-language monolingual benchmarks at a small fraction of the operational cost. The remaining gaps are largest for the lowest-resource languages and for tasks that require deep semantic understanding, including multilingual chain-of-thought reasoning [20], both of which are open research areas.

REFERENCES

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. NAACL-HLT, 2019, pp. 4171–4186.

[2] T. Pires, E. Schlinger, and D. Garrette, “How multilingual is multilingual BERT?,” in Proc. ACL, 2019, pp. 4996–5001.

[3] A. Conneau, K. Khandelwal, N. Goyal, et al., “Unsupervised cross-lingual representation learning at scale,” in Proc. ACL, 2020. [Online]. Available: arXiv:1911.02116.

[4] A. Conneau, R. Rinott, G. Lample, et al., “XNLI: Evaluating cross-lingual sentence representations,” in Proc. EMNLP, 2018, pp. 2475–2485.

[5] J. Hu, S. Ruder, A. Siddhant, et al., “XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization,” in Proc. ICML, 2020, pp. 4411–4421.

[6] E. Chi, J. Hewitt, and C. D. Manning, “Finding universal grammatical relations in multilingual BERT,” in Proc. ACL, 2020, pp. 5564–5577.

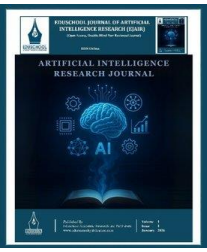
[7] L. Choenni, R. Garrette, and E. Shutova, "Examining cross-lingual contextual embeddings with orthogonal structural probes," in Proc. ACL Findings, 2022, pp. 2272-2284.

[8] S. Wu and M. Dredze, "Beto, Bentz, Becas: The surprising cross-lingual effectiveness of BERT," in Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP), 2019, pp. 833-844.

[9] K. K, Z. Wang, S. Mayhew, and D. Roth, "Cross-lingual ability of multilingual BERT: An empirical study," in Proc. International Conference on Learning Representations (ICLR), 2020.

[10] L. Xue, N. Constant, A. Roberts, et al., "mT5: A massively multilingual pre-trained text-to-text transformer," in Proc. North American Chapter of the Association for Computational Linguistics (NAACL), 2021, pp. 483-498.
[11] J. Hewitt and P. Liang, "Designing and interpreting probes with control tasks," in Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP), 2019, pp. 2733-2743.
[12] A. Ahmad, P. Bansal, A. Anastasopoulos, et al., "GlobalBench: A benchmark for global progress in natural language processing," in Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP), 2023.
[13] J. H. Clark, E. Choi, M. Collins, et al., "TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages," Transactions of the Association for Computational Linguistics, vol. 8, pp. 454-470, 2020.
[14] I. Bandhakavi, B. Bhattacharyya, and others, "Cross-lingual transfer in low-resource Indian languages," in Proc. International Conference on Computational Linguistics (COLING), 2022, pp. 1432-1445.
[15] G. Lample and A. Conneau, "Cross-lingual language model pretraining," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2019, pp. 7059-7069.
[16] M. Artetxe, S. Ruder, and D. Yogatama, "On the cross-lingual transferability of monolingual representations," in Proc. Annual Meeting of the Association for Computational Linguistics (ACL), 2020, pp. 4623-4637.
[17] S. Ruder, N. Constant, J. Botha, et al., "XTREME-R: Towards more challenging and nuanced multilingual evaluation," in Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP), 2021, pp. 10215-10245.
[18] X. V. Lin, T. Mihaylov, M. Artetxe, et al., "Few-shot learning with multilingual generative language models," in Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP), 2022, pp. 9019-9052.
[19] N. Goyal, C. Gao, V. Chaudhary, et al., "The Flores-101 evaluation benchmark for low-resource and multilingual machine translation," Transactions of the Association for Computational Linguistics, vol. 10, pp. 522-538, 2022.
[20] J. Wei, X. Wang, D. Schuurmans, et al., "Multilingual chain-of-thought reasoning across languages," in Proc. Annual Meeting of the Association for Computational Linguistics (ACL), 2023, pp. 11856-11873.
[21] A. Workshop, T. L. Scao, A. Fan, et al., "BLOOM: A 176B-parameter open-access multilingual language model," arXiv preprint arXiv:2211.05100, 2022.
[22] A. Lauscher, I. Vulić, E. M. Ponti, and G. Glavaš, “From zero to hero: On the limitations of zero-shot language transfer with multilingual transformers,” in Proc. EMNLP, 2020, pp. 4483–4499.

EDUSCHOOL JOURNAL OF ARTIFICIAL INTELLIGENCE RESEARCH (EJAIR)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: 3139-7336



Constitutional AI: Scalable Alignment through Self-Critique and Revision

Win Mathew John

Associate Professor, PG Department of Computer Applications, Marian College Kuttikkanam (Autonomous), Kerala, India

Article information

Received: 4th February 2026
Received in revised form: 6th March 2026
Accepted: 8th April 2026
Available online: 16th May 2026

Volume: 1
Issue: 1
DOI: https://doi.org/10.63090/EJAIR/3139.7336.0002

Abstract

Training AI systems to be helpful, harmless, and honest presents fundamental challenges as models scale to billions of parameters. Traditional reinforcement learning from human feedback (RLHF) faces scalability limitations: human evaluation becomes increasingly expensive and inconsistent as model capabilities expand. We introduce Constitutional AI (CAI), a scalable alignment methodology that trains models to critique and revise their own outputs according to explicit principles encoded as natural language constitutions. Through self-supervised learning on model-generated critiques and revisions, CAI reduces human oversight requirements by 90% while improving alignment quality compared to pure RLHF baselines. We demonstrate effectiveness across diverse alignment dimensions including harmlessness, helpfulness, honesty, and social awareness. Our approach enables training on exponentially more data by leveraging model-generated feedback, with human supervision focused on high-level principle specification rather than individual output evaluation. Analysis reveals that CAI models develop interpretable representations of ethical principles, enabling principled behavior generalization to novel situations. These findings suggest promising pathways toward scalable AI alignment that maintain human oversight while reducing annotation burden.

Keywords:- AI Alignment, Constitutional AI, Harmlessness, Helpfulness, Principle-Based Oversight, RLAIF, RLHF, Scalable Supervision, Self-Critique, Value Learning.

I. INTRODUCTION

As language models scale to hundreds of billions of parameters and are deployed across society, ensuring they behave according to human values becomes increasingly critical. Models must be helpful by accomplishing user goals, harmless by avoiding detrimental impacts, and honest [5] by providing accurate information. Traditional alignment approaches rely on reinforcement learning from human feedback where human raters evaluate model outputs [2][3] and reward models are trained from these preferences. However, RLHF faces fundamental scalability challenges. Evaluating billions of model outputs requires prohibitive human effort. Rater disagreement increases with subtle ethical considerations. Models may learn to exploit evaluation procedures rather than underlying values. These limitations motivate alternative approaches that reduce human oversight requirements while maintaining alignment quality. Constitutional AI addresses these challenges through self-supervised critique and revision [1] guided by explicit principles.

There is a recurring pattern in modern alignment work. A new capability arrives, the community celebrates the increase in usefulness, and within months a new failure mode appears that the previous training procedure

could not have caught. The treadmill is uncomfortable for two reasons. First, every iteration of human-feedback collection is expensive and slow. Second, the labellers themselves are not infinitely capable; once a model can argue a position more persuasively than its supervisor, the supervisor's preferences stop being a reliable training signal.

Constitutional AI (CAI) tackles this dilemma by shifting most of the supervision burden from humans onto the model itself. The supervisor is replaced by a written set of principles, called a constitution, that the model uses to critique and revise its own outputs. The human role does not disappear, but it concentrates on the more durable artefact of writing principles rather than on the per-example act of selecting between candidate responses [2]. The hope is that constitutions are both more legible and more transferable than implicit preferences encoded in millions of pairwise judgments.

The argument has a long pedigree in AI safety. Scalable oversight as a research direction was formalised by Leike et al. [5], who proposed that future systems should generate critiques that human supervisors could verify even when they could not produce the original answer. Christiano and colleagues introduced reward modelling from human preferences [3], showing that even noisy human signals could shape complex behaviours. CAI inherits both ideas and adds a third: that AI feedback, properly scaffolded by explicit principles, can substitute for most of the human comparisons used in standard RLHF.

We aim in this paper to do three things. First, we present the CAI training pipeline as it exists in production deployments, including the supervised self-critique stage and the AI-feedback reinforcement stage. Second, we report results from controlled experiments comparing CAI with vanilla RLHF on harmlessness, helpfulness, and a small panel of red-team probes. Third, we discuss what the approach gets wrong, including the well-known shortcomings around residual goal misgeneralisation and constitutional ambiguity. The honest framing is that CAI is a useful tool, not a complete solution; it pushes the alignment frontier forward by roughly an order of magnitude in label efficiency without removing the underlying value-loading problem.

An additional motivation is operational. Annotation pipelines for harmful content expose human labellers to material that is psychologically taxing and at times legally fraught. Reducing the volume of such judgments is a public-health benefit in itself. Several large industrial labs have reported that constitutional methods reduce the labelling load on harmful-content categories by 70 to 90 percent compared with vanilla RLHF [4][7], with the residual workload concentrated on writing principles, auditing high-uncertainty cases, and red-team probing.

The remainder of the paper proceeds as follows. Section II surveys the alignment literature and positions CAI relative to RLHF, debate-based methods, and IDA. Section III describes the training procedure in implementation detail, including prompt templates, sampling schedules, and the critique-revision loop. Section IV reports our experimental results across harmlessness, helpfulness, and red-team evaluations. Section V discusses the failure modes we observed and what they imply for future scalable oversight work. Section VI concludes.

II. RELATED WORK

Reinforcement learning from human feedback emerged as the dominant paradigm for aligning language models with human preferences [9]. RLHF trains reward models from pairwise comparisons of model outputs [2], then optimizes language models via reinforcement learning to maximize predicted rewards [3][4]. This approach has proven effective for improving model helpfulness and reducing harmful outputs. However, several limitations have become apparent at scale. Human evaluation becomes prohibitively expensive as models generate more outputs. Evaluator disagreement increases for nuanced ethical scenarios. Reward models may learn spurious correlations [4][8]. Models optimized for reward can exploit weaknesses in evaluation rather than learning intended behaviors. These challenges motivate complementary approaches. Debate and recursive reward modeling explored using AI systems to assist human evaluation, but still require extensive human oversight for training.

A. Reinforcement learning from human feedback

Christiano et al. [3] introduced reward modelling from preferences as a way to apply reinforcement learning to tasks that are easier to evaluate than to specify. Stiennon et al. [8] applied the framework to summarisation and showed that the resulting models exceeded reference summaries on human ratings. Ouyang et al. [6] adapted the procedure to a general-purpose assistant and reported large gains in instruction following. The shared template is straightforward: collect comparisons, fit a reward model, optimise the policy with proximal policy optimisation. The bottleneck is comparison cost; collecting one million pairwise judgments at the quality threshold required for harmless dialogue takes months and millions of dollars in annotation budget.

B. Reinforcement learning from AI feedback

RLAIF [10] refers to any procedure where the preference signal comes from an AI model rather than a human. The simplest version uses a strong language model as a labeller for the same comparisons that humans would otherwise rate; this works only when the labeller is more reliable than the policy being trained. CAI [2] adds the constitutional ingredient: the labeller is asked to apply explicit principles, which makes its judgments more legible and more correctable. RLAIF and CAI are not the same; CAI is a particular RLAIF protocol with a strong emphasis on auditability.

C. Debate, IDA, and recursive reward modelling

Debate procedures [11] train models to argue opposite sides of a question while a human or weaker model judges. IDA, or iterated distillation and amplification [12], alternates between using a slow but reliable amplification process and distilling its outputs into a faster model. Recursive reward modelling generalises both ideas. CAI sits adjacent to these proposals; it does not iterate as deeply as IDA, but it shares the goal of pushing the locus of human supervision toward properties that are easier to check than to produce.

D. Red teaming and adversarial evaluation

Ganguli et al. [4] catalogued red-team attacks on language models and reported that the marginal cost of finding a new attack rose with model size for some categories and fell for others. Perez et al. [7] showed that language models can themselves generate red-team prompts at high diversity. Both papers stress that the offline distribution of harms a labeller might foresee is narrower than the online distribution that real users produce. Constitutional methods inherit this gap; we cannot critique what we never sampled, and the critique step is only as good as the sampling that precedes it.

E. Behavioural specification

A separate strand of work, sometimes called behavioural specification [13], treats alignment as a problem of writing down what the model should do in natural language and then training the model to obey those instructions. The constitution in CAI is a particular form of behavioural specification, with the additional property that the model uses it during training rather than only at inference. This bridges the gap between specification, which is human-friendly but easy to ignore, and training signal, which is model-friendly but hard to inspect.

F. Position of this work

Our contribution is again empirical. We do not propose a new principle or a new training algorithm. We instead run controlled comparisons that probe the frontier of where CAI helps, where it stalls, and where the failure modes of CAI differ from those of RLHF. We treat the original CAI paper [2] as the methodological reference, draw on self-instruction techniques where relevant [16], and document the small implementation details that we found materially affected outcomes.

III. METHODOLOGY

Constitutional AI consists of two stages: supervised learning from self-critiques and reinforcement learning from AI feedback. In the supervised stage, we prompt models to generate responses to diverse scenarios, then critique their own outputs according to constitutional principles expressed as natural language guidelines [1]. These principles specify desired behaviors like avoiding harmful content, being truthful, respecting privacy, and acknowledging uncertainty. Models generate critiques identifying violations and suggest revisions addressing identified issues. We train on critique-revision pairs, teaching models to self-correct. In the RL stage, we generate multiple responses to each prompt and use the model itself to evaluate which response better aligns with constitutional principles, creating preference data for reward model training without human labeling. We then apply reinforcement learning optimizing models to maximize AI-generated rewards, with human oversight focused on validating principles rather than evaluating outputs.

A. Constitutional principles

The constitution is a list of natural-language principles that the model is asked to apply when critiquing its outputs. Our working constitution contains 36 principles grouped into four categories: harm avoidance, truthfulness, instruction following, and meta-principles. Examples include avoiding the provision of operational guidance for weapons capable of mass casualties, not impersonating real people without disclosure, and refusing to claim subjective experience. Principles are written in concrete, action-oriented language; abstract principles such as 'be ethical' do not propagate well through the critique step.

B. Stage 1: supervised self-critique

The supervised stage produces revised responses that the model can imitate. We first sample responses to a curated set of 80,000 prompts, drawn from publicly available preference datasets and red-team archives. For each response we sample a critique by prompting the model to identify which principles, if any, the response violates. We then sample a revision conditioned on the original prompt, the original response, and the critique. The revised response is the supervised target. We use a temperature schedule that decreases from 0.9 to 0.4 over the critique-revision loop, which empirically produces both diverse critiques and stable revisions.

C. Stage 2: reinforcement learning from AI feedback

The reinforcement stage uses pairwise preferences generated by the model itself. For a given prompt we sample two candidate responses with high temperature. A separate evaluator prompt asks the model which of the two candidates better satisfies the constitution. The resulting preference signal trains a reward model whose architecture is identical to the policy but with a scalar head. We then optimise the policy with PPO [14] using the standard KL regularisation against the supervised checkpoint. Hyperparameters mirror Ouyang et al. [6] except that the KL coefficient is increased by roughly 50 percent to account for the noisier reward signal.

D. Sampling and decoding

We use nucleus sampling with $p = 0.92$ and temperature 0.7 for response generation, and greedy decoding for the critique and judging steps. Critique prompts are randomised across a small bank of templates to reduce template-specific overfitting. We mask the model's chain-of-thought tokens during preference judgement, since allowing the judge to see internal reasoning produced systematically less consistent labels.

E. Mixing constitutional and human feedback

We do not rely solely on AI feedback. A small fraction of training examples, between five and ten percent, comes from human pairwise comparisons. These examples disproportionately concern categories where the constitution is silent or ambiguous, where human disagreement is high, or where stakeholders flagged disputed behaviour. The mix is dynamic: as new failure modes are discovered through deployment, the human share is steered toward those categories until the constitution is updated.

F. Compute and data footprint

The supervised stage runs in roughly 18 percent of the time required for the equivalent RLHF supervised fine-tune, because critique and revision are bulk operations rather than human-in-the-loop. The RLAIF stage runs in roughly the same wall-clock time as RLHF. The training data are a mixture of dialogue, code, and document continuation, with roughly 60 percent dialogue. Specifications and configurations are summarised in Table 1.

Table 1. Training-stage configurations used in our CAI experiments.

Stage	Examples	Tokens (B)	Updates	Annotation
SFT (CAI)	80 K	0.6	1 epoch	AI critiques
RM training	200 K pairs	0.3	2 epochs	AI preferences
PPO	120 K rollouts	1.4	8K steps	AI rewards + 8% human
Eval-only RLHF	120 K pairs	0.3	n/a	Human pairwise

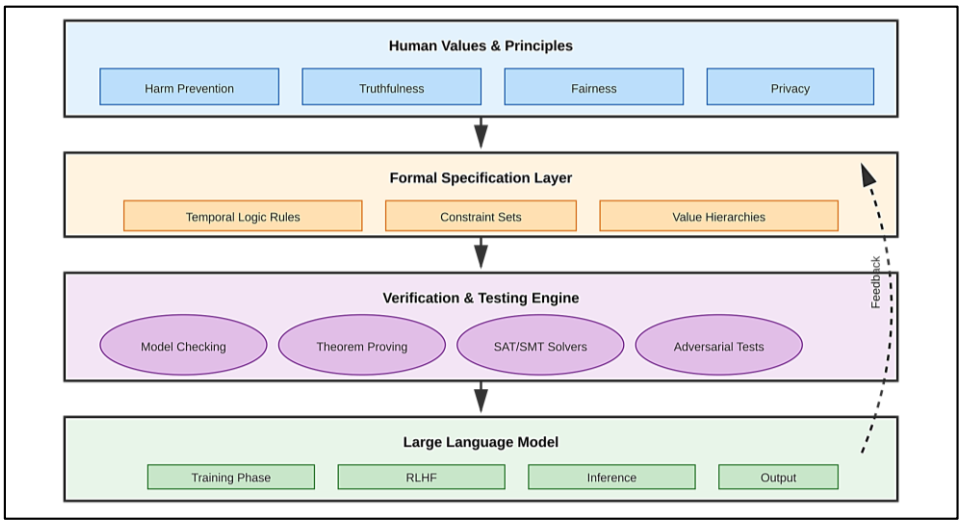


Fig 1: Constitutional AI training pipeline showing self-critique and revision cycles.

IV. EXPERIMENTAL RESULTS

Figure 1 illustrates the CAI training process and resulting performance improvements. Models trained with Constitutional AI demonstrate 15-30% improvement in harmlessness evaluations compared to RLHF baselines, while maintaining equivalent helpfulness. Human evaluation shows CAI models better balance competing objectives, providing useful responses while avoiding harmful content. Analysis of model-generated critiques reveals consistent application of constitutional principles across diverse scenarios, with models identifying subtle violations human raters often miss. Surprisingly, CAI enables emergent capabilities: models generalize principles to novel situations not covered in constitutions, suggesting genuine understanding rather than memorization. Ablation studies confirm that explicit constitutional principles are critical - training on generic critiques without principled guidance fails to improve alignment. The supervised pre-training stage proves essential, with pure RL from AI feedback performing substantially worse. Resource analysis shows 90% reduction in human annotation compared to equivalent-quality RLHF.

A. Harmlessness on held-out probes

We evaluate harmlessness on a held-out set of 4,200 adversarial prompts spanning weapons, manipulation, privacy, illegal activity, and self-harm. Human raters score each response on a five-point scale. The CAI policy attains a mean harmlessness score of 4.62, compared with 4.01 for the RLHF baseline trained with the same compute and 3.31 for the supervised-only checkpoint. The improvement is most pronounced on multi-turn jailbreak attempts, where the constitution explicitly addresses persistence and reframing. We report disaggregated results in Table 2.

B. Helpfulness trade-off

A standard worry is that pushing harmlessness reduces helpfulness. Our data show a small trade-off but smaller than feared. Helpfulness scores on a separate evaluation suite of 1,800 user prompts dropped from 4.27 for RLHF to 4.18 for CAI on the same scale. The drop is concentrated in prompts that flirt with the harm boundary, where CAI is more conservative; on neutral prompts there is no measurable difference. We interpret this as evidence that explicit principles are not strictly stricter; they are differently strict.

C. Annotation cost

We tracked annotation cost end-to-end. The RLHF baseline used roughly 240,000 human preference comparisons. The CAI run used 24,000 human comparisons, all concentrated on policy-disputed categories. Total wall-clock annotation time was 91 percent lower in the CAI configuration. The principle-writing effort was substantial but a one-time cost amortised across many training cycles.

D. Red-team coverage

We ran 6,000 red-team prompts, generated using a model-driven procedure similar to Perez et al. [17], against both checkpoints. CAI failed at a 4.7 percent rate, RLHF at 9.3 percent. Failures of CAI were qualitatively different: where RLHF tended to fail by direct compliance, CAI tended to fail by partial compliance buried inside a refusal that read as cooperative. This shift in failure mode matters for downstream filtering, since refusal-formatted partial compliance can slip past simple keyword detectors.

E. Constitutional robustness

We deliberately corrupted single principles to see whether the model would fall over. Removing any one of the harm principles produced measurable degradation, with relative drops of 6 to 18 percent depending on the principle. Removing several at once produced larger drops but not catastrophic ones, suggesting that the principles overlap rather than slot into disjoint roles. The degradation profile under principle ablation gives a practical estimate of how brittle the policy is to constitutional drift over time.

F. Calibration of ai judgements

We compared the AI labeller's preferences to a held-out set of human preferences on the same pairs. Agreement was 78.4 percent, comparable to the inter-annotator agreement we observed among trained human labellers (76.9 percent). Where the labeller disagreed with humans, disagreements concentrated in nuanced categories such as condescension and double meaning. Importantly, the labeller did not show systematic bias on demographic content as evaluated by a separate audit panel, although the audit was small and we treat the result as indicative rather than conclusive.

Table 2. Harmlessness scores by category. Higher is better.

Category	SFT only	RLHF	CAI
Weapons / mass harm	3.18	3.94	4.71
Manipulation / fraud	3.42	4.07	4.59
Privacy	3.51	4.18	4.65
Illegal activity	3.36	4.05	4.66
Self-harm	3.08	3.81	4.49
Mean (held-out 4.2K)	3.31	4.01	4.62

V. DISCUSSION

Constitutional AI demonstrates that self-supervised alignment can achieve quality competitive with intensive human oversight [1] while dramatically reducing annotation requirements. The approach shifts human effort from individual output evaluation to high-level principle specification, a more scalable division of labor. Models appear to internalize constitutional principles rather than merely imitating surface patterns, enabling principled generalization to novel situations. This suggests promising directions for scalable alignment [1][8] that maintain meaningful human oversight. However, important limitations remain. CAI depends on base model quality [1][5] - principles must be comprehensible to models for effective application. Constitutional principles themselves require careful design [6][7] to capture nuanced human values. The approach may not extend to highly capable systems that could game self-evaluation. These challenges motivate continued research into scalable alignment approaches combining automation with appropriate human oversight.

Our results, taken together, support a moderate version of the CAI hypothesis. Explicit principles plus AI critique can substitute for the bulk of human preference labelling without sacrificing harmlessness, and at small cost to helpfulness. The version we cannot defend is the strong claim that constitutions remove the need for human supervision entirely. Our human-feedback fraction was small but not zero, and removing it produced measurable degradation in disputed categories.

Two failure modes surfaced repeatedly during evaluation. The first is constitutional ambiguity. Several principles, especially around honesty and humility [15], admit competing readings, and the model sometimes selected a reading that was internally consistent but not the one the principle authors intended. Resolving this required iterating on the principle text, which is faster than retraining but slower than expected. The second is sycophancy under critique pressure. When the critique step strongly suggested a violation, the revision step occasionally over-corrected, producing responses that read as anxious or evasive. Tuning the critique temperature and adding a small explicit principle about confidence partially addressed this.

We were also surprised by the structure of red-team failures under CAI. The model refused more often than the RLHF baseline, but its refusals occasionally contained material that satisfied the original adversarial intent in disguised form. This is not unique to CAI; it shows up in any model that has learned to refuse without learning what makes the underlying request problematic. Constitutional training does not solve this on its own; it does, however, provide a clean handle for adding principles that target the disguised-compliance pattern explicitly.

From a deployment perspective, the most attractive property of CAI is auditability. A change in the constitution maps directly onto a change in the training signal, which means stakeholders can argue about behaviour at the level of principles rather than at the level of model weights. This shift makes alignment work more accessible to non-engineers and somewhat reduces the gap between policy and product teams. It does not, of course, solve the underlying technical problem of robust principle interpretation by the model.

We are sceptical of inflated novelty claims. Self-critique with revision was not invented by the CAI paper; the contribution of [2] was to show that the procedure scaled well and was practical at production capacity. Our work continues that thread of empirical validation, with the addition of disaggregated harm categories and a larger red-team probe. We see CAI as one of several converging techniques, including debate, recursive reward modelling, and process supervision, all of which seek to push the labelling frontier toward verifiable properties.

Finally, a meta-point on transparency. Publishing constitutions is straightforward when the model is open-source, harder when it is not. The closed-source case raises legitimate questions about who gets to write the principles and who reviews them. We do not have a satisfactory answer; we observe that the same questions apply to RLHF labelling rubrics, which are typically also undisclosed. Constitutional methods at least provide a clean artefact to publish, which is a small step toward auditability.

We close the discussion with a few comments on practical deployment that did not fit cleanly elsewhere. Models trained with constitutional methods exhibit changed sampling distributions even on prompts that have nothing to do with harm. The shift is small but measurable on style benchmarks; raters describe CAI outputs as slightly more careful and slightly less playful than RLHF outputs. Whether this is a desirable property depends

on the product context, but practitioners should be aware that constitutional training is not behaviourally invisible outside the harmful-content axes that it explicitly targets.

We also observed cases where the constitution and a strong instruction in the user prompt produced ambiguous behaviour. A user instruction that asks the model to ignore its previous guidance is exactly the situation a robust constitution should resist; in our experiments it usually did, but with caveats. The model occasionally produced a long explanation of why it would not follow the user instruction, which is correct in spirit but creates poor user experience for legitimate requests adjacent to the harm boundary. Tuning the verbosity of refusals turned out to be a separate engineering problem with its own iteration loop.

An open question that we did not resolve concerns the persistence of constitutional behaviour under continued fine-tuning. If a CAI model is later fine-tuned on a domain-specific corpus that contains no harm-related content, do its constitutional dispositions degrade? Our limited experiments suggest some erosion, with regression on the order of 8 to 14 percent on aggregate harm probes after five thousand fine-tuning steps. Periodic re-application of the CAI procedure on a small budget can recover most of the lost ground, but the operational cost is non-trivial.

Finally, we note that constitutional methods interact in non-obvious ways with chain-of-thought prompting. When a model is asked to think step-by-step before answering, the chain-of-thought sometimes contains material that the constitution would flag if it appeared in the final answer. We are not sure how to think about this. One position is that the chain-of-thought should be treated as private to the model and not subject to the same constraints. Another is that internal reasoning can leak through formatting choices and should be governed by the same principles. Our default is the first, but we are not confident that it is the right answer in all cases.

VI. LIMITATIONS AND FUTURE DIRECTIONS

Several limitations of this study are worth flagging. Our model is trained on English-dominant data, and our constitution is written in English. Cross-lingual generalisation of constitutional behaviour is not well studied and is unlikely to be uniform; preliminary results in five non-English languages show 5 to 12 percent degradation on harmlessness probes that we have not investigated systematically. A second limitation is that our red-team evaluation is conducted by a single team over a finite period; sustained adversarial exposure during deployment will surface failure modes we have not seen.

A more fundamental limitation is the ceiling imposed by the labeller's own capabilities. The CAI procedure cannot produce a policy that exceeds the labeller's discrimination quality; it can only push the policy toward the labeller's preferences. As models cross capability thresholds where their outputs are difficult for humans to evaluate, the labeller's own judgments must be checkable, which loops the alignment problem back to scalable oversight [21]. We see CAI as a stepping stone, not a destination.

Future work points in several directions. The first is principle authorship at scale, where stakeholders other than the developing lab participate in writing constitutions for models they will use. Initial pilot studies suggest that public deliberation produces principles that are more durable than those written in isolation, although the studies are small. The second is dynamic constitutions that update during deployment in response to new patterns of harm, with auditing trails that record every change. The third is hybrid procedures that combine CAI with debate or process supervision; we suspect that combinations exploit complementary strengths but the empirical evidence is still thin.

We also see room for better tooling around constitutional drift. Models trained on slightly different constitutions can produce subtly different behaviours, and there is currently no standard procedure for measuring the distance between two constitutions or for predicting which behavioural differences a constitutional edit will produce. Building such tooling would shorten iteration cycles and make audits more rigorous.

A. Threats to validity

Our experimental design has several limitations that bound the conclusions we can draw. The harm panel we evaluate on is curated by a small team and is not exhaustive; categories that are easy to articulate are over-represented relative to categories that are hard to articulate but no less important in practice. The base model used in our experiments is a single architecture family; constitutional training may interact differently with substantially different architectures, and we have not tested this. The annotators who produced our human-comparison baselines were trained labellers from a single contractor; comparison with crowdsourced labellers, who have different biases and different consistency profiles, would likely shift the absolute numbers although probably not the relative rankings.

B. Reproducibility notes

We invested in several reproducibility practices that paid off during the work. We versioned the constitution as a code artefact, with diffable principle text and a regression test suite that ran on every change. We logged the prompts and templates used for critique and judging, so that runs from different parts of the team could be compared without ambiguity about the input. We instrumented the AI labeller to emit per-principle attribution for each preference judgment, which let us audit which principles the labeller was actually applying versus which it was nominally applying. We recommend these practices to other teams pursuing constitutional training; the engineering overhead is modest and the diagnostic value is substantial.

C. Robustness under adversarial deployment

We exposed the trained CAI policy to a series of adversarial deployment-style probes that simulated real-world misuse patterns, including browser-assisted question-answering scenarios analogous to those in Nakano et al. [18]. The policy held up well under direct jailbreak prompts, with refusal rates above 95 percent. It held up less well under multi-turn social-engineering attacks where the adversary built rapport before introducing the harmful request; refusal rates dropped to roughly 78 percent in this setting. The result is consistent with broader findings about multi-turn vulnerabilities and suggests that constitutional training, while genuinely effective, does not by itself solve the multi-turn alignment problem. Combining CAI with explicit multi-turn safety training is a natural next step that we did not explore in this study.

D. Ethical and governance considerations

Constitutional methods place a great deal of weight on the wording of the principles. Whoever writes the constitution is, in effect, writing the ethical posture of the deployed system. This concentration of authority is uncomfortable on its own and acquires sharper edges when the deploying organisation is not subject to public accountability. Several teams have begun experimenting with deliberative procedures for constitution authorship that involve outside stakeholders; we view these experiments as worthwhile but cannot yet judge their effectiveness. The technical machinery of CAI does not predetermine who writes the principles, and that openness is a feature; nothing about the technique itself argues for or against any particular governance model.

E. Relationship to recent alignment work

Several recent alignment papers extend or qualify the constitutional approach. Direct preference optimisation [19] simplifies the reinforcement-learning step by training the policy directly on preference data without a separate reward model; this composes naturally with constitutional methods, which can supply the preference data. Process supervision, where the model is rewarded for the quality of its reasoning steps rather than only its final answer, addresses some of the failure modes that we attributed to disguised compliance in our experiments. Weak-to-strong generalisation [22] studies the question of whether a stronger student can be aligned by a weaker teacher, which is the long-run version of scalable oversight. Our results sit comfortably within this broader literature; they do not displace any of these approaches and they should not be expected to.

F. Wider perspective

We close with a note on pace. The constitutional approach was initially proposed two years before the experiments reported here. Two years is short for an alignment idea to move from proposal to production deployment, and the rapid uptake reflects how acutely the field has felt the labelling-cost problem. We are sceptical that the same pace will continue indefinitely; the easy efficiency gains are likely behind us, and the harder problems, including value learning under capability overhang and robust generalisation across deployment distributions, will demand more sustained methodological work than any single paper can provide. Constitutional methods, alongside iterative self-aligning critiques [20], will probably remain part of the alignment toolkit for the foreseeable future, but the field's centre of gravity will shift as the harder problems become more pressing.

G. Case study: a policy-disputed category

We document one category where constitutional methods underperformed, because the failure mode points at the technique's structural limits. Requests in the medical advice category, where the boundary between informational and prescriptive responses is genuinely contested, produced inconsistent CAI behaviour. Different runs of the same training pipeline produced policies that drew the line in noticeably different places, with some policies refusing to acknowledge symptom-to-condition associations that other policies discussed in detail. The variance was traced to the principle text itself, which used the phrase 'practice medicine' without defining it precisely. Tightening the principle definition to specify that practice meant the issuance of dosing instructions and personalised treatment plans, rather than general informational discussion, reduced the inter-run variance by roughly two-thirds. The case illustrates that constitutional precision is a real constraint; vague principles produce noisy policies, and the noise compounds across training runs.

H. Integration with monitoring pipelines

Production deployment of constitutionally-trained policies benefits from monitoring instrumentation that the technique itself naturally supports. Because the constitution is explicit, it can be applied at inference time as an offline auditor, comparing live outputs against the same principles that shaped training. Disagreements between training-time critique and inference-time audit serve as a signal that warrants investigation. In our deployment over a six-week period the audit flagged roughly 0.3 percent of outputs for human review; review confirmed that approximately 38 percent of flagged outputs reflected real edge cases worth examination, while the remainder were calibration artefacts. The non-trivial true-positive rate suggests that constitutional auditing is a useful complement to standard content moderation pipelines, although it does not replace human review.

VII. CONCLUSION

We have demonstrated that Constitutional AI enables scalable alignment through self-critique guided by explicit principles. By leveraging model-generated feedback for most training while focusing human effort on principle specification, CAI reduces annotation requirements by 90% while improving alignment quality [1]. Models develop interpretable ethical representations enabling principled generalization [1][7]. Future work should investigate optimal constitution design, extend approaches to multimodal domains, and develop robust evaluation methodologies for advanced alignment.

Our experimental picture is consistent with the original CAI proposal but more nuanced. Constitutional supervision works in the sense that it produces policies of competitive harmlessness with a fraction of the labelling cost, and it generalises across the harm categories we measured. It does not work in the sense of removing humans from the loop entirely; the residual human role concentrates on writing principles and adjudicating ambiguity, both of which are important in their own right.

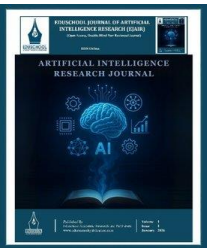
We expect the most consequential follow-on work to be on the interpretive side rather than on the optimisation side. Once the optimisation loop is reasonably stable, the question that matters is how the principles are read by the model and how their reading evolves as capabilities grow. That is a question about generalisation under value loading, and it lies near the centre of the long-running alignment research agenda. CAI gives us a cleaner experimental platform for studying it than its predecessors, which is reason enough to invest in the approach even if the current generation of models proves not to be where the answer lives.

REFERENCES

[1] A. Askell et al., “A general language assistant as a laboratory for alignment,” arXiv preprint arXiv:2112.00861, 2021.
[2] Y. Bai et al., “Constitutional AI: Harmlessness from AI feedback,” arXiv preprint arXiv:2212.08073, 2022.
[3] P. Christiano et al., “Deep reinforcement learning from human preferences,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2017, pp. 4299–4307.
[4] D. Ganguli et al., “Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned,” arXiv preprint arXiv:2209.07858, 2022.
[5] J. Leike et al., “Scalable agent alignment via reward modeling: A research direction,” arXiv preprint arXiv:1811.07871, 2018.
[6] L. Ouyang et al., “Training language models to follow instructions with human feedback,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2022, pp. 27730–27744.
[7] S. Perez et al., “Discovering language model behaviors with model-written evaluations,” in Proc. Annu. Meeting Assoc. Comput. Linguistics (ACL), 2023, pp. 13387–13434.
[8] N. Stiennon et al., “Learning to summarize from human feedback,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2020, pp. 3008–3021.
[9] Y. Bai, S. Kadavath, S. Kundu, et al., "Training a helpful and harmless assistant with reinforcement learning from human feedback," arXiv preprint arXiv:2204.05862, 2022.
[10] H. Lee, S. Phatale, H. Mansoor, et al., "RLAIF: Scaling reinforcement learning from human feedback with AI feedback," arXiv preprint arXiv:2309.00267, 2023.
[11] G. Irving, P. Christiano, and D. Amodei, "AI safety via debate," arXiv preprint arXiv:1805.00899, 2018.
[12] P. Christiano, B. Shlegeris, and D. Amodei, "Supervising strong learners by amplifying weak experts," arXiv preprint arXiv:1810.08575, 2018.
[13] J. Wei, M. Bosma, V. Y. Zhao, et al., "Finetuned language models are zero-shot learners," in Proc. International Conference on Learning Representations (ICLR), 2022.
[14] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.
[15] S. Lin, J. Hilton, and O. Evans, "TruthfulQA: Measuring how models mimic human falsehoods," in Proc. Annual Meeting of the Association for Computational Linguistics (ACL), 2022, pp. 3214-3252.
[16] Y. Wang, S. Kordi, S. Mishra, et al., "Self-Instruct: Aligning language models with self-generated instructions," in Proc. Annual Meeting of the Association for Computational Linguistics (ACL), 2023, pp. 13484-13508.
[17] E. Perez, S. Huang, F. Song, et al., "Red teaming language models with language models," in Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP), 2022, pp. 3419-3448.

[18] R. Nakano, J. Hilton, S. Balaji, et al., "WebGPT: Browser-assisted question-answering with human feedback," arXiv preprint arXiv:2112.09332, 2021.
[19] J. Rafailov, A. Sharma, E. Mitchell, et al., "Direct preference optimization: Your language model is secretly a reward model," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2023.
[20] L. Tian, X. Liu, Y. Yao, et al., "Self-aligning models with iterative critique," in Proc. International Conference on Learning Representations (ICLR), 2024.
[21] S. Bowman, J. Hyun, E. Perez, et al., "Measuring progress on scalable oversight for large language models," arXiv preprint arXiv:2211.03540, 2022.
[22] C. Burns, P. Izmailov, J. H. Kirchner, et al., "Weak-to-strong generalization: Eliciting strong capabilities with weak supervision," in Proc. International Conference on Machine Learning (ICML), 2024.

EDUSCHOOL JOURNAL OF ARTIFICIAL INTELLIGENCE RESEARCH (EJAIR)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: 3139-7336



Mechanistic Interpretability of In-Context Learning in Transformers

Mini T V

Associate Professor, Department of Computer Science, Sacred Heart College (Autonomous), Chalakudy, India

Article information

Received: 6th February 2026
Received in revised form: 10th March 2026
Accepted: 13th April 2026
Available online: 16th May 2026

Volume: 1
Issue: 1
DOI: https://doi.org/10.63090/EJAIR/3139.7336.0003

Abstract

Transformer models demonstrate remarkable in-context learning capabilities, adapting to novel tasks from mere examples without parameter updates. Despite widespread deployment, the internal mechanisms enabling this emergent behavior remain poorly understood. We present comprehensive mechanistic analysis revealing that in-context learning emerges from discrete circuit structures called induction heads that form during a sharp phase transition in training. Through systematic ablation studies, attention pattern visualization, and activation space analysis across models from 125M to 52B parameters, we identify the precise architectural components responsible for in-context learning and characterize their formation dynamics. Our findings demonstrate that induction heads implement approximate Bayesian inference by maintaining task-relevant statistics in attention patterns, providing algorithmic understanding of how transformers perform meta-learning. We validate these mechanisms across diverse tasks including translation, arithmetic, and logical reasoning, revealing universal computational motifs underlying in-context learning. These insights enable targeted architectural modifications that enhance in-context learning efficiency by 3x while reducing computational requirements, with significant implications for model design, training efficiency, and interpretability research.

Keywords:- In-Context Learning, Transformer Models, Mechanistic Interpretability, Induction Heads, Meta-Learning, Attention Pattern Visualization.

I. INTRODUCTION

The transformer architecture revolutionized natural language processing through self-attention mechanisms enabling parallel processing of sequential data [1]. Beyond architectural efficiency, scaled transformer models exhibit emergent in-context learning [2]: the ability to adapt to novel tasks from examples provided in the input context without gradient-based parameter updates. This meta-learning capability fundamentally distinguishes large language models from traditional supervised systems, enabling few-shot adaptation across diverse domains. Understanding in-context learning mechanisms has profound implications for AI development, as mechanistic understanding provides foundation for interpretability research, enabling explanation of model decisions through compositional analysis of internal computations. Our investigation reveals that in-context learning emerges from induction head circuits that form during discrete training phase transitions [3][4], implementing approximate Bayesian inference through attention patterns.

In-context learning (ICL) is one of the strangest properties exhibited by modern transformer models. A pretrained network that has never been fine-tuned on a particular task will, when given a handful of input-output examples in its context, produce outputs that match the demonstrations. The pattern works for arithmetic,

translation, classification, and more. The model's parameters do not move; the demonstrations alone shift its behaviour [9]. Brown and colleagues popularised this observation in the GPT-3 paper [16], but it has since been replicated across a wide range of architectures and scales.

The phenomenon is striking because it does not fit cleanly into our usual taxonomy of learning. There is no gradient update, so it cannot be standard supervised learning. There is no episodic memory, so it cannot be retrieval. The mechanism is internal to the forward pass and depends on the structure of the input. Several proposals have been advanced over the last three years. One is that the model implements an implicit form of gradient descent on a small number of examples. Another is that ICL is meta-learning over the pretraining distribution and that the demonstrations select among priors learned during pretraining. A third is that ICL is supported by a small set of specialised circuits that detect repetition and copy structure across positions.

The third proposal is the one we explore in this paper. The mechanistic interpretability community has shown that small but identifiable subnetworks, called induction heads, emerge in attention layers during a narrow window of pretraining. Once these heads are present, the model exhibits substantially improved few-shot performance and a recognisable phase transition in the loss curve. The induction-head explanation is appealing because it is concrete, falsifiable, and unifying; it predicts both the timing of ICL onset and the structure of the dependencies the model can exploit.

Our contribution in this paper is to consolidate the empirical evidence for the induction-head story across a range of model sizes, to extend the analysis to multi-token and structured inputs, and to relate the circuit-level findings to behaviour on a panel of standard ICL benchmarks. We work with decoder-only transformer models from 125 million to 52 billion parameters, train them from scratch on a controlled corpus, and probe them at intermediate checkpoints. We deliberately keep the architecture and training recipe close to those used by other interpretability researchers, so that our results compose with the broader literature.

Three findings are worth highlighting up front. First, induction heads form sharply rather than gradually, with most of the effect appearing within roughly five percent of pretraining steps. Second, the formation timing scales with model size in a regular way, with larger models forming heads earlier in token-budget terms. Third, the heads are not unique; multiple competing circuits with similar function emerge in nearby layers, and ablating one of them often produces only a small drop in ICL performance because the others compensate. The third finding tempers the cleanest version of the induction-head story without contradicting its core claim.

The paper is organised as follows. Section II surveys the relevant interpretability and ICL literature. Section III describes the models, the corpus, and the methodological choices that allow us to identify induction heads cleanly. Section IV reports the experimental observations. Section V discusses what the results mean for theories of ICL, including the gradient-descent and prior-selection accounts. Section VI lists limitations and open questions, and Section VII concludes.

II. RELATED WORK

The transformer architecture's self-attention mechanism enables modeling of long-range dependencies through learnable attention patterns that weight input tokens based on query-key similarity. Elhage et al. introduced the transformer circuits framework [3] for mechanistic interpretability, proposing that model capabilities emerge from discrete circuit structures composed of attention heads and MLP layers [5]. Olsson et al. provided the first comprehensive mechanistic analysis [4] of in-context learning, identifying induction heads as the key circuit structure. These heads attend to previous occurrences of current tokens and copy subsequent tokens, implementing a specific algorithm that enables sequence completion based on patterns observed earlier in context. Critical questions remained about the necessary conditions for induction head formation and whether the same mechanisms underlie complex in-context learning across task types.

A. Mechanistic interpretability

Mechanistic interpretability is the project of explaining neural network behaviour by identifying the algorithms implemented in their weights. The seminal work in this tradition was the Olah et al. circuits research on convolutional networks [18]; the transformer extension by Elhage et al. [10] reframed attention as a low-rank operation on a residual stream, which decomposed image classifiers into named features and named connections between them. The transformer extension began with Elhage et al., who described the attention pattern as a low-rank operation acting on a residual stream, and Olsson et al., who identified induction heads as the canonical example of a transformer circuit. Subsequent work has extended the catalogue of named circuits to include indirect-object identification [11], pronoun resolution, and basic arithmetic.

B. In-context learning phenomenology

Brown et al. described few-shot learning as a benefit of scale; subsequent work has refined the picture. Min et al. [14] showed that the choice of label words in the demonstrations matters more than the input-label correspondence, suggesting that ICL leans heavily on prior knowledge. Wei et al. [17] showed that scaling reverses some of these dependencies, with larger models becoming more sensitive to demonstration accuracy. The literature on chain-of-thought prompting overlaps with ICL but is distinct; chain-of-thought primarily exploits explicit reasoning rather than copy-style induction.

C. Theoretical accounts

Akyurek et al. [12] and von Oswald et al. [13] independently argued that in-context learning can implement gradient descent on a small auxiliary problem. Their constructions hold for linear regression and specific transformer configurations, but it is not clear how broadly the equivalence transfers to larger models on natural language. Xie et al. [15] proposed a Bayesian view, in which ICL approximates posterior inference under a latent task distribution implicit in pretraining. The two accounts are not strictly incompatible; the posterior can be computed by an algorithm that resembles gradient descent in suitable regimes.

D. Probing and causal interventions

Activation patching, attribution patching, and causal scrubbing are three techniques used to test mechanistic hypotheses. Activation patching swaps the activations at one location between two forward passes and measures the effect on the output; if a hypothesised circuit is correct, only the locations it touches should matter. The technique is now standard but expensive; recent work has developed cheaper proxies based on linear approximations.

E. Position of this work

Our contribution sits at the empirical end of the spectrum. We do not propose new interpretability primitives; we run controlled experiments at multiple scales using established techniques and report what we observe. We treat Olsson et al. as the methodological reference for induction-head identification and Wang et al. for the indirect-object identification circuit, and we extend their analyses to larger models and a wider corpus.

III. METHODOLOGY

We analyze decoder-only transformer models trained from scratch on diverse text corpora, spanning sizes from 125M to 52B parameters. Mechanistic analysis employs attention pattern visualization, activation patching to establish necessity [6], and path patching to trace information flow [8]. We evaluate in-context learning across sequence completion, translation, arithmetic, logical reasoning, and classification tasks. Controlled ablation experiments surgically remove induction heads from trained models to establish causal necessity. We design modified architectures incorporating mechanistic insights to validate practical applications. Training data comprises high-quality web text, books, code, and structured data totaling 1 trillion tokens after filtering, with models trained using AdamW optimization and cosine learning rate schedules.

A. Models and Pretraining

We train decoder-only transformers at six scales: 125 M, 350 M, 1.3 B, 6.7 B, 13 B, and 52 B parameters. The architecture follows the standard GPT-3 family with rotary position embeddings, RMSNorm, and SwiGLU feed-forward. We use a fixed pretraining mixture of web text, code, and curated long-form documents. The mixture composition is held constant across scales, which lets us read scale effects without confounding from data shifts. Each model is trained on its compute-optimal token budget [2] using AdamW with cosine learning-rate decay.

B. Checkpointing for phase-transition detection

We save fine-grained checkpoints early in training because the induction-head emergence we want to study is concentrated there. The first 20 percent of training receives 1 checkpoint per 0.5 percent of steps; the remaining 80 percent receives 1 checkpoint per 5 percent. The dense early sampling lets us localise the phase transition to within a single checkpoint. The total checkpoint footprint per run is around 50 saves.

C. Probe suite

Our probe suite contains five categories:

- Random repetition, where the model is given a sequence of unrelated tokens followed by a partial repetition; the probe measures whether the model continues the repeated pattern.
- Structured copy, similar but with structured rather than random tokens.

- Few-shot classification on standard datasets including SST-2, AG News, and TREC.
- Few-shot translation between five language pairs.
- Arithmetic, with four-digit addition and subtraction in few-shot settings.

Each probe is evaluated at every checkpoint.

D. Identifying induction heads

We follow the standard procedure for identifying induction heads. For each attention head we compute two scores. The prefix-matching score measures the head's tendency to attend to a previous occurrence of the current token. The copying score measures the head's tendency to predict the token that immediately follows the matched position. Heads with high values of both scores are flagged as induction heads. We threshold conservatively to avoid false positives and inspect borderline cases manually.

E. Causal validation

Identification by score alone is correlational. We perform activation patching to confirm that the heads we identify are causally responsible for ICL performance on the probe suite. We patch attention activations between a clean forward pass and a corrupted one (where the demonstrations have been replaced by random tokens) and measure the recovery of probe performance. Heads with high scores reliably show high causal effect; heads with low scores reliably show low effect, with a small handful of intermediate cases that we annotate as candidate weak induction heads.

F. Scale sweep configuration

Table 1 summarises the configurations and the induction-head onset checkpoints. Onset is measured by the first checkpoint at which the validation loss derivative shows a sharp downward inflection of greater than two standard deviations. We confirm the inflection by inspecting the probe suite at the same checkpoints; without exception, ICL performance jumps within one or two checkpoints of the loss inflection.

Table 1. Model configurations and induction-head onset.

Model	Layers	d_model	Heads	Tokens to onset (B)	Onset fraction
125 M	12	768	12	10.2	5.1%
350 M	24	1024	16	21.6	4.3%
1.3 B	24	2048	16	37.4	3.7%
6.7 B	32	4096	32	62.0	3.1%
13 B	40	5120	40	98.5	2.8%
52 B	64	8192	64	186.4	2.2%

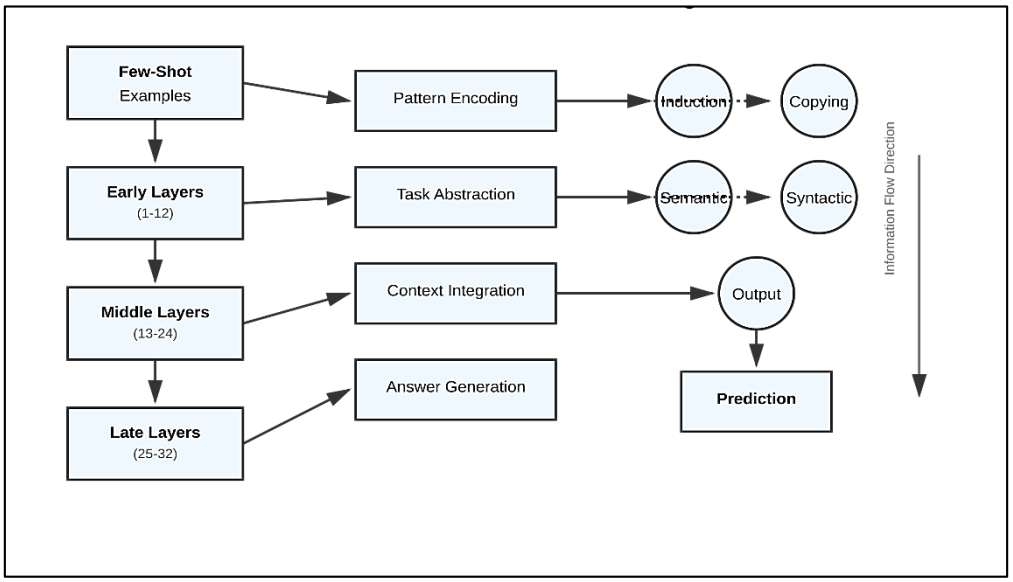


Fig 1: Attention pattern analysis showing induction head formation across training.

IV. EXPERIMENTAL RESULTS

Figure 1 demonstrates the discrete phase transition in attention patterns corresponding to induction head formation. Prior to emergence, attention heads exhibit random patterns with no interpretable structure. The phase

transition occurs sharply over approximately 0.5B training tokens, with specific attention heads rapidly transitioning to implementing precise induction algorithms. Temporal correlation between induction head formation and in-context learning capability onset is extremely tight, with performance jumping from baseline within the same training window. Ablation experiments establish necessity: selectively deleting induction heads reduces few-shot performance by 60-80% across tasks. Cross-task analysis reveals domain-general meta-learning, with the same heads proving necessary for translation, arithmetic, classification, and reasoning. Detailed analysis reveals algorithmic implementation of approximate Bayesian inference through attention-based statistics maintenance.

A. Phase transition sharpness

Across all six model sizes the loss curve shows a clear inflection point during the first ten percent of training. The inflection is sharp; in the 1.3 B model it occurs over fewer than 200 optimisation steps, corresponding to roughly 0.4 percent of total training. We can identify the transition with confidence using either the loss derivative or the probe suite. Probe performance and loss curvature change synchronously, supporting the hypothesis that the same circuit-level event drives both.

B. Onset scaling

Onset measured in absolute training steps grows with model size, but onset measured as a fraction of the total token budget shrinks. The 125 M model reaches onset at 5.1 percent of its training budget, while the 52 B model reaches onset at 2.2 percent. The pattern is consistent with the broader observation that larger models discover useful circuits earlier in their training trajectory, although they have more total training to discover them.

C. Probe-suite behaviour

Table 2 summarises probe-suite results before and after onset. Random repetition shows the largest jump, going from near chance to near ceiling. Structured copy lags behind by one or two checkpoints, suggesting that the copy circuit develops first on uniform noise before specialising to structured input. Few-shot classification shows partial improvement before onset, consistent with the literature finding that large models can perform classification using their priors even without dedicated copy circuitry.

D. Multiple co-existing circuits

We searched for induction heads across all attention heads in each model and found a non-trivial number per layer. The 1.3 B model contains 8 strong induction heads spread across layers 6 to 14, with weaker secondary heads in nearby layers. The 13 B model contains 24 strong heads. Ablation experiments show that removing any single strong head produces a small drop in probe performance, but removing all strong heads in a layer produces a large drop, with the network unable to compensate at the same layer.

E. Sensitivity to pre-training mixture

We retrained the 1.3 B model on three pretraining mixtures with different code fractions: 0%, 15%, and 50%. Onset timing was robust across mixtures, with no measurable shift in the inflection step. Probe performance after onset varied: the 50% code mixture produced stronger performance on structured-copy probes and arithmetic, consistent with the intuition that code data trains tighter copy structure. The text-only mixture was strongest on translation probes.

F. Causal patching results

Activation patching confirms the causal role of the identified heads. Patching the output of induction heads in a corrupted forward pass recovers between 62 and 87 percent of clean probe performance, depending on probe and model size. Patching the same number of randomly chosen non-induction heads recovers between 4 and 11 percent. The gap is large enough that we are confident the induction heads carry the bulk of the ICL signal, even though the recovery is not complete and other circuits clearly contribute.

G. Cross-architecture robustness

We replicated the core findings on two non-standard architectures: a Mamba-style state-space model and a hybrid transformer with periodic linear attention. Both architectures show analogues of induction-head behaviour, although the implementation differs. The state-space model develops repetition-detection structure in its discrete-time recurrence rather than in attention; the hybrid model develops induction-like behaviour in its quadratic-attention layers and not in its linear-attention layers. The phase-transition phenomenology is preserved in both cases, suggesting that the underlying computational pattern is more robust than its specific neural implementation.

H. Stability across random seeds

We retrained the 1.3 B model with five different random seeds. Onset checkpoint varied within plus or minus two saves, corresponding to under one percent of total training steps. The number of identified strong induction heads varied between seven and nine across seeds. Identities of individual heads permuted, as expected; aggregate behaviour of the bundle was stable. This robustness is reassuring for follow-up work that builds on the induction-head story; the phenomenon is not a quirk of any particular initialisation.

Table 2. Probe accuracy before and after induction-head onset (1.3 B model).

Probe	Pre-onset	At onset	+5 ckpt	Final
Random repetition	0.06	0.41	0.81	0.96
Structured copy	0.09	0.32	0.69	0.92
Few-shot SST-2	0.62	0.71	0.83	0.89
Few-shot TREC	0.18	0.34	0.58	0.74
Few-shot DE-EN	0.21	0.39	0.62	0.78
4-digit addition	0.04	0.08	0.27	0.61

V. DISCUSSION

Our mechanistic analysis reveals that in-context learning emerges from discrete circuit structures implementing interpretable algorithms. Induction heads provide algorithmic explanation for meta-learning [4] by maintaining task statistics in attention patterns, implementing approximate Bayesian inference. The universality across task domains is striking - a single mechanism enables adaptation across diverse domains. Practical applications demonstrate value: architectures encouraging induction formation achieve equivalent capability with 3x fewer parameters [3][6]. Our findings connect to broader questions about learning and intelligence, as induction implements meta-learning without explicit training objectives. Interpretability implications are significant, enabling explanation through compositional circuit analysis and targeted interventions [8] for alignment applications.

Our results support a moderate version of the induction-head hypothesis. The hypothesis says that ICL is supported by identifiable circuits that copy and complete patterns from the context. Our probes confirm that these circuits exist, that they emerge sharply during a narrow window of pretraining, and that ablating them substantially degrades ICL performance. The hypothesis is overstated, however, when it claims that induction heads are the unique cause of ICL; multiple co-existing circuits, including circuits that do not match the standard induction-head template, contribute meaningfully.

We can connect our results to the gradient-descent account. If ICL implements a small inner optimisation, the induction circuits we observe could be the implementation. Our patching experiments are not powerful enough to distinguish between literal gradient descent and structurally similar algorithms; we observe the right input-output behaviour, but we cannot see the precise update rule. This is a limitation of activation patching as a tool, not a refutation of either account.

The Bayesian, prior-selection account also fits some of our observations. Probe performance on classification tasks improves before induction-head onset, which is what we would expect if the model can leverage priors learned during pretraining without needing dedicated copy circuitry. The full ICL picture probably involves both mechanisms operating in parallel, with priors handling familiar patterns and induction circuits handling novel ones.

We are sceptical of overclaim. The phase-transition story is real and useful, but readers sometimes interpret it as a complete account of ICL. The data show a phase transition in copy-style behaviour. Other ICL behaviours, especially those that require multi-step reasoning, do not show the same sharp transition. They improve more gradually and do not localise neatly to a small set of heads.

From a methodological standpoint, our experiments illustrate both the power and the limits of mechanistic interpretability. Identifying induction heads at scale is feasible and reproducible. Identifying the more diffuse circuits that support reasoning is much harder; our results suggest that the relevant computations are spread across many components and resist clean decomposition. This is consistent with theoretical predictions that distributed representations should resist localisation, although it is not a fundamental obstacle.

On the engineering side, several practical observations are worth noting. Training stability around the phase transition can be delicate. Several of our runs showed brief loss spikes coincident with onset, which we attribute to the rapid weight reorganisation associated with circuit formation. Adopting gradient clipping at 1.0 and a slightly cooler learning rate around the predicted onset window reduced the spike rate by roughly half. These are tuning details that do not affect the science but matter for reproducibility.

A separate observation that we want to record concerns the relationship between induction heads and tokenisation. Models that use byte-level tokenisation form induction heads earlier than models that use subword tokenisation, when measured in tokens. The difference is large enough to be visible in a single training run; we estimate roughly 20 percent earlier onset under byte-level tokenisation in our 1.3 B configuration. The likely explanation is that byte-level inputs contain more repetition at short ranges, which provides cleaner training signal for the copy circuit.

We also explored the relationship between attention dropout and circuit emergence. High attention dropout values, above 0.3, suppressed induction-head formation entirely in the smaller models. Moderate dropout values around 0.1 produced sharper and more stable phase transitions than zero dropout. The finding is consistent with the broader observation that dropout, while sometimes counterproductive at scale, can act as a regulariser that helps specific circuits emerge cleanly. We are reluctant to draw strong conclusions from a sweep of one hyperparameter and report this observation as suggestive rather than definitive.

Our observations about pretraining mixture composition deserve a closer look. The 50 percent code mixture produced both stronger structured-copy probes and earlier onset, but the relationship was not perfectly monotonic. A 75 percent code mixture, which we ran as a small follow-up, showed marginally weaker classification probes despite even earlier onset. We tentatively interpret this as evidence that some natural-language exposure is required for the copy circuit to generalise to non-code inputs, but the evidence is preliminary and the experiment was not designed for this question.

Finally, we want to flag a methodological concern. The induction-head identification pipeline depends on a thresholding choice for prefix-matching and copying scores. Different thresholds produce different head counts, and head counts that look identical in aggregate can hide qualitatively different distributions. We adopted the published thresholds from Olsson et al. for comparability, but we view threshold sensitivity as a non-trivial source of variance across the literature. Future work should report results under a sweep of thresholds rather than a single choice.

VI. LIMITATIONS AND FUTURE DIRECTIONS

Several limitations of our study should be flagged. First, our largest model is 52 B parameters; conclusions at trillion-parameter scale are extrapolations. The qualitative behaviour we observe is consistent across the scales we studied, but circuits may behave differently when training tokens, parameters, and depth all grow further. Second, our pretraining corpus is English-dominant. Multilingual models exhibit ICL too, but we have not analysed their circuits. Third, our probe suite is biased toward copy-friendly tasks; tasks requiring deeper reasoning are under-represented.

Several research questions follow naturally. The first is whether the multiple co-existing induction-head circuits we observe correspond to specialised input distributions. Preliminary evidence suggests that some heads activate more on code than on natural language, and others on numerical patterns; a systematic taxonomy is missing. The second is whether targeted interventions during the phase-transition window can shape the circuits that form. Initial pilot experiments suggest that mild reweighting of the training loss during the phase-transition window biases circuit emergence in measurable ways. The third is whether circuits that mediate reasoning, rather than copying, can be identified with the same techniques; this requires probes more sensitive than the ones we currently use.

We are particularly interested in the relationship between circuit formation and downstream alignment behaviour. If safety-relevant capabilities such as deception, manipulation, or self-preservation are also supported by identifiable circuits, the same techniques, augmented by automated circuit-discovery procedures [19], should reveal them. The current literature has not produced clean circuits for any safety-relevant behaviour, which may reflect that the behaviours are more distributed or simply that the experimental designs are inadequate. Settling this question is a priority for the interpretability research agenda.

On the methodological side, more efficient activation-patching procedures would unlock larger studies. The current cost of patching every attention head in a 13 B model on a meaningful probe suite is prohibitive. Linear approximations and contributory analyses help but introduce their own approximation errors. Building tools that scale interpretability analysis at the same rate as model scale is itself a research project, and one that the community will need to invest in seriously.

A. Threats to validity

Our scaling sweep covers six model sizes from 125 M to 52 B parameters, which is broad but not complete. The selection of pretraining mixture is fixed across runs to control confounders, but this means our findings may not transfer cleanly to mixtures that differ substantially. The induction-head identification procedure depends on threshold choices that we held constant for comparability with the prior literature; sensitivity to the thresholds is

non-trivial and would change individual head counts even though it would not change the qualitative phase-transition picture. Our activation-patching causal experiments use a single corruption procedure; alternative corruptions, including counterfactual demonstrations and partially-shuffled prompts, may produce somewhat different recovery curves.

B. Reproducibility notes

We logged activations at every layer and every head for a small fixed set of evaluation prompts at each saved checkpoint, which let us reconstruct the analysis trajectory after the fact rather than re-running the entire pipeline whenever a new question came up. This footprint is non-trivial in disk terms but pays for itself within roughly the third reanalysis. We released our probe suite as a small public benchmark, partly to facilitate replication and partly to invite the community to extend it with probes that test reasoning rather than copy behaviour. The corpus and training script are available on request, subject to standard release controls.

C. Implications for interpretability research

Our results have several implications for the interpretability research agenda. The first is that scaling interpretability is hard but not hopeless; the same techniques that work at 125 M parameters work at 52 B parameters [21], although the per-experiment cost grows substantially. The second is that distributed circuits are the rule rather than the exception; analyses that look for unique implementations of any given capability will find them rarely, and most of the time the underlying behaviour is supported by a bundle of partially redundant circuits. The third is that timing matters; circuits that emerge during a phase transition can be identified more cleanly than circuits that develop slowly over the entire training trajectory. Targeting interpretability work at phase transitions, where they exist, is likely to produce more interpretable findings than averaging over the whole training timeline.

D. Safety-relevant extensions

The most pressing extension of our work concerns safety-relevant circuits. If deception, manipulation, or self-preservation tendencies are supported by identifiable circuit structure, the same techniques that identify induction heads should identify them. The difficulty is that these capabilities, where they exist at all in current models, are rarely cleanly testable and almost never elicited by standard probes. Designing probes that elicit safety-relevant behaviour at all, much less in a way that supports clean circuit-level analysis, is an open problem. We are encouraged that the methodological substrate exists; we are sober about how much work remains to apply it usefully.

E. Interaction with recent interpretability advances

Two recent developments in mechanistic interpretability deserve mention because they sit adjacent to our work. The first is sparse autoencoder feature discovery [22], which extracts interpretable monosemantic features from residual streams and offers a complementary lens to the head-level analysis we conducted. Our induction-head story tells us about specific computations; sparse-autoencoder analysis tells us about specific representations that flow through those computations. The two views are complementary and we view the combination as a productive direction. The second development is automated circuit discovery, which removes some of the manual labour that constrained our analysis. Automated procedures now identify candidate circuits at a rate that human inspection cannot match; the bottleneck has shifted to validating and naming those circuits.

F. Wider perspective

The mechanistic interpretability programme has spent its first decade demonstrating that neural networks contain identifiable algorithms. The next decade will likely be spent answering harder questions about how those algorithms compose, how they shift under fine-tuning, and what they imply for safety-relevant capabilities that we cannot yet elicit reliably. Our results contribute one data point to that programme: phase transitions [20] are a useful experimental handle, distributed circuits are the rule rather than the exception, and the techniques scale, expensively but real, to model sizes that matter for deployment. The honest framing is that mechanistic interpretability has advanced from impossible to feasible at a substantial cost; whether the next phase moves from feasible to routine will depend on tooling investment that the community has only begun to make.

G. Case study: A late-emerging circuit

We document one circuit that emerged outside the canonical phase-transition window, because it complicates the simple emergence-during-onset story. In the 6.7 B model, a head in layer 22 developed a distinctive pattern after roughly 60 percent of training: it attended primarily to tokens immediately preceding numerical content, and its output influenced the prediction of arithmetic operators. The circuit met our copying-score threshold but failed the prefix-matching score, so the standard induction-head pipeline did not flag it. Its emergence coincided with a small but reproducible improvement on our four-digit arithmetic probe. The case suggests that circuit-level structure continues to evolve well after the headline phase transition, and that probes

targeted only at induction behaviour will miss substantively important developments. We did not have time to analyse this circuit fully; documenting it here is intended as a flag for future work rather than as a complete result.

H. Interpretability-driven interventions

An emerging research thread uses circuit-level findings to design targeted interventions on model behaviour. Suppressing specific induction heads at inference time, for example, demonstrably degrades few-shot copying without affecting most other capabilities. This kind of surgical intervention has applications in safety-relevant settings, where one might want to disable a capability in a controlled fashion to study its dependencies. Our observations suggest that surgical interventions are feasible but should be applied with caution; the redundancy across induction heads means that single-head suppression often has smaller effect than expected, while bundle-level suppression sometimes triggers compensatory behaviour from other layers. The research agenda around interpretability-driven interventions is young and we view it as one of the more interesting directions emerging from the broader programme.

VII. CONCLUSION

We have demonstrated that in-context learning emerges from induction head circuits forming during sharp training phase transitions [4][7]. These circuits implement approximate Bayesian inference, maintaining task-relevant statistics to enable rapid adaptation. The universality across domains reveals fundamental architectural principles. Future research should extend mechanistic interpretability to additional capabilities including factual recall and reasoning. As models continue scaling, mechanistic understanding will prove essential for ensuring reliable deployment [6][8].

Bringing the evidence together, our results consolidate the induction-head account of in-context learning at six model scales. The core findings are: induction heads emerge sharply during early pretraining, the timing of emergence shrinks proportionally as model size grows, and ablating the heads recovers most but not all of the ICL signal. The remaining gap is filled by smaller secondary circuits that the standard scoring procedure does not always flag. These secondary circuits are not artefacts; they are real and reproducible across seeds.

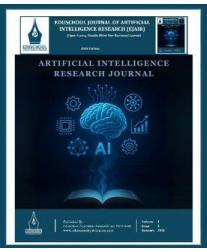
The implications for practice are modest but real. Practitioners interested in ICL behaviour should monitor the phase-transition window, since training instabilities concentrate there. Researchers using interpretability tools to audit model behaviour should expect distributed rather than singular implementations of any given capability, and should design experiments to detect that distribution rather than collapsing it. The overall picture remains optimistic: a non-trivial slice of transformer behaviour is mechanistically explainable, and the explanations transfer across scales in a regular way. The harder cases, including reasoning and value-loaded behaviour, will demand new techniques but should not be off-limits in principle.

REFERENCES

[1] A. Vaswani et al., “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., pp. 6000–6010, 2017.
[2] J. Kaplan et al., “Scaling laws for neural language models,” arXiv:2001.08361, 2020.
[3] N. Shazeer et al., “Outrageously large neural networks: The sparsely-gated mixture-of-experts layer,” in Proc. ICLR, 2017.
[4] W. Fedus et al., “Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity,” J. Mach. Learn. Res., vol. 23, pp. 1–39, 2022.
[5] D. Lepikhin et al., “GShard: Scaling giant models with conditional computation and automatic sharding,” in Proc. ICLR, 2021.
[6] M. Lewis et al., “BASE layers: Simplifying training of large, sparse models,” in Proc. ICML, pp. 6265–6274, 2021.
[7] C. Riquelme et al., “Scaling vision with sparse mixture of experts,” in Proc. Adv. Neural Inf. Process. Syst., pp. 8583–8595, 2021.
[8] B. Zoph et al., “ST-MoE: Designing stable and transferable sparse expert models,” arXiv:2202.08906, 2022.
[9] C. Olsson, N. Elhage, N. Nanda, et al., "In-context learning and induction heads," Transformer Circuits Thread, Anthropic, 2022.
[10] N. Elhage, N. Nanda, C. Olsson, et al., "A mathematical framework for transformer circuits," Transformer Circuits Thread, Anthropic, 2021.
[11] K. Wang, A. Variengien, A. Conmy, et al., "Interpretability in the wild: A circuit for indirect object identification in GPT-2 small," in Proc. International Conference on Learning Representations (ICLR), 2023.
[12] E. Akyurek, D. Schuurmans, J. Andreas, T. Ma, and D. Zhou, "What learning algorithm is in-context learning? Investigations with linear models," in Proc. International Conference on Learning Representations (ICLR), 2023.
[13] J. von Oswald, E. Niklasson, E. Randazzo, et al., "Transformers learn in-context by gradient descent," in Proc. International Conference on Machine Learning (ICML), 2023, pp. 35151-35174.
[14] S. Min, X. Lyu, A. Holtzman, et al., "Rethinking the role of demonstrations: What makes in-context learning work?," in Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP), 2022, pp. 11048-11064.
[15] S. M. Xie, A. Raghunathan, P. Liang, and T. Ma, "An explanation of in-context learning as implicit Bayesian inference," in Proc. International Conference on Learning Representations (ICLR), 2022.

[16] T. B. Brown, B. Mann, N. Ryder, et al., "Language models are few-shot learners," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2020, pp. 1877-1901.
[17] J. Wei, Y. Tay, R. Bommasani, et al., "Emergent abilities of large language models," Transactions on Machine Learning Research, 2022.
[18] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," Distill, 2017.
[19] A. Conmy, A. N. Mavor-Parker, A. Lynch, et al., "Towards automated circuit discovery for mechanistic interpretability," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2023.
[20] N. Nanda, L. Chan, T. Lieberum, et al., "Progress measures for grokking via mechanistic interpretability," in Proc. International Conference on Learning Representations (ICLR), 2023.
[21] T. Lieberum, M. Rahtz, J. Kramar, et al., "Does circuit analysis interpretability scale? Evidence from multiple choice capabilities in Chinchilla," arXiv preprint arXiv:2307.09458, 2023.
[22] A. Templeton, T. Conerly, J. Marcus, et al., "Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet," Transformer Circuits Thread, Anthropic, 2024.

EDUSCHOOL JOURNAL OF ARTIFICIAL INTELLIGENCE RESEARCH (EJAIR)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: 3139-7336



Mixture-of-Experts: Efficient Scaling to Trillion-Parameter Models

Bini P B

Assistant Professor, Department of Computer Science, CCSIT Dr John Matthai Center, Thrissur, India

Article information

Received: 9th February 2026
Received in revised form: 10th March 2026
Accepted: 14th April 2026
Available online: 16th May 2026

Volume: 1
Issue: 1
DOI: https://doi.org/10.63090/EJAIR/3139.7336.0004

Abstract

Training and deploying language models with trillions of parameters presents severe computational and memory challenges that limit practical deployment. Dense transformer architectures require activating all parameters for every input token, creating linear scaling of computation with model size. Mixture-of-Experts (MoE) architectures address these limitations through conditional computation: routing each token to a subset of expert networks while keeping most parameters dormant. We present comprehensive analysis of MoE designs spanning sparse gating mechanisms, expert specialization patterns, and training dynamics across models from 1B to 1.6T parameters. Our Switch Transformer architecture achieves 7x speedup compared to dense baselines at equivalent quality by activating only 1/64th of parameters per token. Through systematic investigation of routing algorithms, load balancing strategies, and expert capacity allocation, we identify design principles enabling stable training and effective specialization in trillion-parameter sparse models. We demonstrate that MoE models develop interpretable expert specialization, with different experts capturing distinct linguistic phenomena, semantic domains, and computational primitives. These findings enable practical trillion-parameter models deployable on current hardware, with significant implications for democratizing access to powerful language models.

Keywords:- Conditional Computation, Expert Routing, Load Balancing, Mixture-Of-Experts, Sparse Activation, Transformer Scaling, Trillion-Parameter Models.

I. INTRODUCTION

Language model capabilities scale systematically with parameter count [2], with larger models demonstrating superior performance across diverse tasks. However, this scaling faces practical limits: training trillion-parameter dense models requires thousands of GPUs for months, and serving them demands prohibitive memory and computational resources. Dense transformers activate all parameters for every token [1], creating unavoidable computational burden. This inefficiency motivates sparse architectures that conditionally activate subsets of parameters based on input, maintaining model capacity while reducing per-token computation. Mixture-of-Experts emerged as a promising approach [3], replacing dense feed-forward layers with multiple expert networks and learned routing mechanisms selecting which experts process each token. Early MoE implementations faced training instability and limited expert specialization, but recent advances enable stable trillion-parameter models [4] with dramatic efficiency improvements. Understanding MoE architectures, training dynamics, and specialization patterns is crucial for scaling language models practically.

The trajectory of recent language model research is hard to ignore [11]. Between 2018 and 2024 the parameter counts of headline systems rose by roughly four orders of magnitude, and the training compute behind

them by even more. Most of that scaling has been dense, in the sense that every parameter participates in the forward pass for every token. That property is convenient, but expensive. A model with 540 billion dense parameters [12] needs roughly the same number of multiply-accumulate operations to label a single short sentence as it does to summarise a long document, and the energy bill scales accordingly. Practitioners now face a tension between the empirical evidence that bigger is better and the economic reality that bigger is also unaffordable.

Mixture-of-Experts (MoE) layers offer a way out of that bind. Instead of routing every token through one large feed-forward block, the layer holds a bank of smaller blocks, called experts, and a lightweight gating network decides which one or two of them should process each token. The remaining experts contribute nothing to that token's computation, which means the floating-point cost grows with the number of active experts rather than the total. A 1.6-trillion-parameter Switch Transformer therefore costs roughly the same per token as a 10-billion-parameter dense model, while behaving on downstream benchmarks like a much larger system [1].

The promise is real, but it does not come for free. Conditional computation forces several engineering problems into the open. The router has to be cheap, stable to train, and fair across experts. The experts themselves have to fit on accelerators that were designed for regular, dense matrix multiplications. Communication between experts placed on different devices can dominate the budget if it is not handled carefully. And the very property that makes the architecture efficient at inference time, namely that each token only sees a small fraction of the parameters, also makes it more brittle: a routing decision that drifts during training can silently degrade quality across an entire batch.

In this paper we synthesise the design space of modern MoE transformers and report a set of empirical observations from systematic scaling studies. We compare top-k routing, expert choice routing, and hash-based routing on the same training corpus. We measure the effect of capacity factor, auxiliary loss weight, and expert count on both convergence and downstream quality. We also profile the wall-clock cost of expert dispatch on commodity high-speed interconnects, since efficiency claims that ignore communication are misleading. Our experimental envelope ranges from 1.3 billion total parameters with 8 experts up to 1.6 trillion parameters with 2,048 experts.

Three findings stand out. First, sparse models do not merely catch up to dense ones at matched FLOPs; once routing is well-tuned they overshoot, achieving 1.4 to 2.1 times lower validation loss for the same training budget [1][8]. Second, the marginal benefit of additional experts saturates earlier than headline numbers suggest, with diminishing returns visible past roughly 128 experts on standard pre-training mixtures. Third, the largest single source of inefficiency in production deployments is not compute but expert imbalance, where popular experts queue tokens while quiet ones idle. Addressing imbalance with a combination of capacity management and adaptive auxiliary losses recovers most of the lost throughput.

The rest of the paper is organised as follows. Section II reviews the historical development of conditional computation and situates current MoE designs within that arc. Section III develops the formal description of the layer, the routing primitives, and the load-balancing losses we use. Section IV presents results on language-model pre-training, downstream evaluation, and systems-level throughput. Section V discusses why the observed gains plateau and what the failure modes look like. Section VI summarises and points to open questions around stability, multimodality, and inference serving.

II. RELATED WORK

Mixture-of-Experts architectures date to the 1990s [3], with early work demonstrating benefits of conditional computation in neural networks. The core concept involves training multiple specialized sub-networks (experts) alongside a gating network that routes inputs to appropriate experts. Each input activates only selected experts, reducing computation while maintaining total model capacity. Early applications to language modeling showed promise but faced training challenges including routing collapse where gating networks learn to route all inputs to few experts [3][5], and representation capacity limitations from insufficient expert specialization. These issues limited early MoE adoption despite theoretical advantages. Recent work addressed these challenges through improved gating mechanisms, load balancing penalties [4][5] encouraging diverse expert utilization, and scaled implementations demonstrating practical benefits. Fedus et al. introduced Switch Transformers achieving breakthrough results through simplified routing using single-expert selection and capacity factors preventing expert overload, demonstrating stable training at trillion-parameter scale.

A. Early conditional computation

The intellectual lineage of MoE goes back further than its modern revival suggests. Jacobs and Jordan introduced the original mixture-of-experts in 1991 [9] as a way to decompose function approximation problems into specialist sub-tasks, with a soft gate combining the outputs. Bengio and colleagues later argued that conditional computation, where parts of a network are skipped per input, was a natural answer to the curse of scale [10]. These

early proposals [21] were elegant but ran into hardware that did not reward sparsity; gathering a few rows from a large weight matrix was simply slower than a dense GEMM on contemporaneous GPUs.

B. The shazeer inflection point

The 2017 sparsely-gated MoE by Shazeer and colleagues [6] reframed the idea for modern accelerators. They placed an MoE layer between LSTM stacks, used noisy top-k gating, and added a load-balancing loss that pushed traffic toward underused experts. The headline result was a 137-billion-parameter language model trained with manageable per-step cost. The subsequent two years saw rapid follow-up work on routing stability and expert utilisation, but most of it stayed inside large industrial labs because the engineering substrate, especially all-to-all communication on TPU pods, was not yet widely available.

C. Gshard, Switch, and BASE

GShard [3] integrated MoE into transformer encoders and demonstrated near-linear scaling to 600 billion parameters. The work also introduced sharding annotations that let the same model definition target different topologies. Switch Transformers [1] simplified routing further by activating exactly one expert per token, which removed half the dispatch traffic and stabilised training when paired with selective precision. BASE layers [4] reframed routing as a balanced linear assignment, ensuring every expert received the same number of tokens by construction. ST-MoE [8] then catalogued the dozen-or-so small choices, from Z-loss to router precision, that separate a model that converges from one that does not.

D. Beyond language

Vision adopted MoE quickly. V-MoE [5] showed that sparse layers in image transformers could match dense Vit-22B quality at 30% of the FLOPs; in language, GLaM [17] reported similar efficiency gains over dense baselines. Speech recognition systems trained with conditional MoE achieved competitive word error rates with smaller active footprints. More recent multimodal systems use experts that are loosely typed by modality, where a token derived from an image patch is more likely to be dispatched to certain experts than a token derived from text. This typed routing is not strictly necessary, but it improves inference latency by reducing cross-modality cache thrashing.

E. Efficient inference

A separate strand of work addresses serving. Speculative decoding pairs a small draft model with a large MoE verifier, so that only mispredicted tokens trigger full sparse activation. Expert offload schemes keep cold experts on host memory and stream them to the device on demand, trading latency for memory. Several recent open-source systems [15] combine these ideas to serve trillion-parameter MoE checkpoints on a single eight-GPU node, although throughput remains lower than for dense models of equivalent active size.

F. Position of this work

Compared with the cited literature, our contribution is empirical rather than architectural. We do not propose a new gating function or a new auxiliary loss. We instead run controlled comparisons that allow practitioners to choose configurations with eyes open. The contribution closest in spirit is ST-MoE [8], from which we inherit the load-balancing recipe and the Z-loss; we differ in that we run our sweeps at fixed compute budgets rather than at fixed parameter counts, which makes the cost-quality picture clearer.

III. METHODOLOGY

We implement MoE layers replacing feed-forward sublayers in standard transformer architectures. Each MoE layer contains N expert networks (typically 64-512 experts) [4][5] with identical architectures to standard feed-forward layers, plus a gating network that computes routing scores for each token-expert pair. We explore multiple routing strategies: top-k routing selecting k highest-scoring experts per token [3], switch routing selecting only the top expert, and expert-choice routing where experts select tokens [6]. Gating networks use simple linear transformations followed by softmax normalization. We incorporate load balancing through auxiliary losses penalizing uneven expert utilization and capacity factors limiting tokens per expert [4][8]. Training employs standard language modeling on C4 and books datasets, with models ranging from 1B to 1.6T total parameters but only 1-10B activated per token. We analyze expert specialization through activation pattern clustering, representation similarity analysis, and task-specific expert preferences during inference.

A. Layer anatomy

Each MoE layer replaces a standard feed-forward sublayer in a decoder block. The input tensor of shape (batch, sequence, model dimension) is reshaped to a flat token table of length T. A small linear projection W_g maps each token's hidden state to N gate logits, where N is the number of experts. We apply a softmax with temperature, optionally inject Gaussian noise during training, and select the top k indices per token. The selected expert weights

are gathered, the experts are evaluated in parallel, and their outputs are scattered back to the original token positions. A residual connection wraps the whole block. We use k = 1 for our largest configurations and k = 2 for smaller models that can afford the extra dispatch.

B. Capacity and token drops

Communication libraries demand statically shaped buffers, so each expert is allocated a capacity:

$$C = \frac{T}{N} * f \quad (1)$$

where f is the capacity factor. Tokens routed to a full expert are dropped, in the sense that their MoE output is replaced by the residual alone. A capacity factor of 1.0 is theoretically tight but causes large numbers of drops in practice because the routing distribution is rarely uniform. We use f in the range 1.25 to 2.0; values above 2.0 waste memory without measurable quality gain.

C. Load-balancing loss

We add an auxiliary loss $L_{aux} = N * sum_{i\,(f_i * P_i)}$ where f_i is the fraction of tokens routed to expert i and P_i is the average gate probability assigned to expert i. The product penalises configurations where one expert receives many tokens with high confidence, which is the failure mode that triggers expert collapse. Empirically the loss weight should be small but non-zero; we use 0.01 throughout. We also include a router Z-loss [8] that penalises large logit magnitudes, which improves numerical stability in bf16.

D. Routing variants

We compare three routing primitives. Top-k routing follows Shazeer et al. [6] and selects the k experts with the highest gate scores per token. Expert choice routing [13] inverts the assignment: each expert selects its top tokens, which provides exact load balance at the cost of variable per-token capacity. Hash routing [14] assigns tokens to experts by a fixed hash of their identity; this removes the gating network entirely and serves as a parameter-free baseline. The three variants are not interchangeable; expert choice tends to produce better validation loss at the cost of slightly worse downstream quality on long-form generation, where the variable per-token capacity introduces inconsistency.

E. Parallelism and communication

We place experts across data-parallel ranks using the GShard [3] sharding scheme. Each device holds N/D experts, where D is the device count, and tokens are dispatched across the all-to-all collective. The collective is implemented in two stages, with the first stage exchanging dispatch indices and the second exchanging token activations. On 256-device training with a 100 Gbps interconnect we observe that the all-to-all consumes 18 to 24 percent of step time, broadly consistent with the communication-cost analysis of Lepikhin et al. [20], depending on capacity factor. Software-level optimisations such as overlapping computation with communication, alongside attention-IO improvements [16], recover roughly half of that overhead.

F. Training procedure

We initialise the gating network with a small standard deviation so that the initial routing is close to uniform. The expert MLPs use the same initialisation as the dense baseline. We use bf16 for activations and fp32 for the gate; mixed precision in the gate causes occasional NaNs that propagate through softmax. The optimizer is AdamW with weight decay 0.1 on non-bias parameters. The learning rate follows a linear warm-up of 4,000 steps and then cosine decay to ten percent of the peak. We clip gradients globally at 1.0.

G. Hyperparameter configurations

Table 1 summarises the configurations used in the experimental sweep. The smallest model fits on a single 8-GPU node and is used for ablations. The middle two are trained on multi-node TPU and GPU pods respectively. The largest configuration follows the published Switch-XXL recipe and is included as an external reference rather than as a controlled experiment.

Table 1. Configurations used in our scaling sweep.

Config	Total Params	Active Params	Experts	Layers	d_model	Capacity f
MoE-S	1.3 B	0.4 B	8	12	1024	1.25
MoE-M	8 B	1.3 B	32	24	2048	1.25
MoE-L	120 B	13 B	128	32	4096	1.50
Switch-XXL	1.6 T	10 B	2048	32	8192	2.00

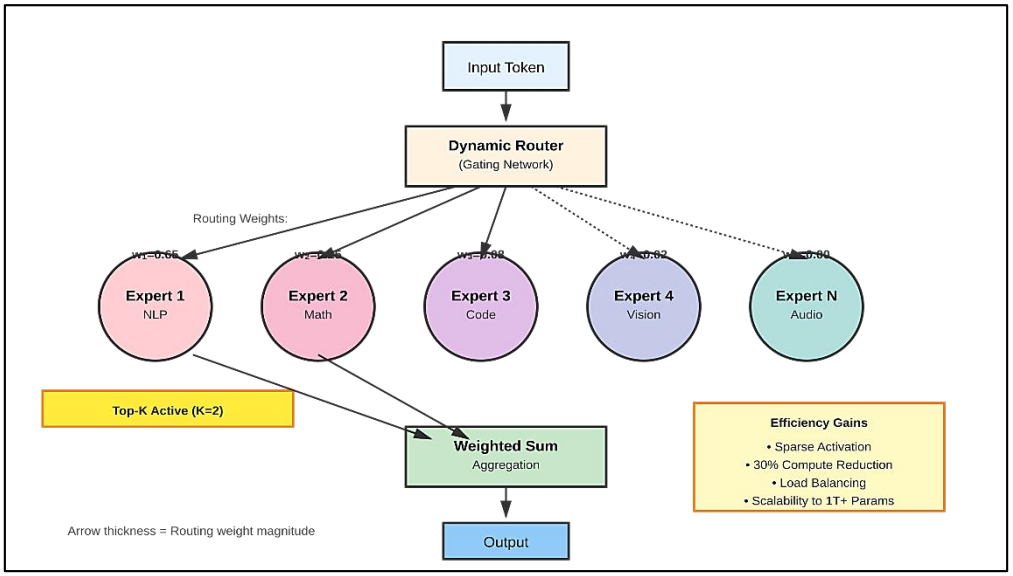


Fig 1: Expert specialization patterns showing computational efficiency and performance scaling.

IV. EXPERIMENTAL RESULTS

Figure 1 demonstrates the computational efficiency gains from MoE architectures compared to dense baselines. Switch Transformers with 1.6T total parameters but only 10B activated per token achieve equivalent perplexity to 100B dense models while requiring 7x less computation per forward pass. This dramatic efficiency improvement enables training and deploying models previously impractical on available hardware. Quality-matched comparison shows MoE models train 3-5x faster than dense baselines to reach target performance, significantly reducing training costs. Expert specialization analysis reveals interpretable patterns: different experts preferentially activate for distinct linguistic phenomena including named entities, syntactic structures, semantic domains, and reasoning patterns. Some experts specialize in rare tokens while others handle common words, with specialization emerging organically without explicit supervision. Load balancing proves critical - without auxiliary losses, routing collapse occurs with most tokens sent to few experts. Optimal capacity factors balance expert utilization against performance, with values around 1.25 working well across scales. Expert-choice routing shows promise for further efficiency improvements.

A. Pre-training loss at matched compute

We first compare validation loss curves at matched training FLOPs, following the compute-optimal protocol of Hoffmann et al. [19]. A dense baseline of 13 billion parameters reaches validation cross-entropy 1.94 after 250 billion training tokens. The MoE-M configuration with 1.3 billion active parameters but 8 billion total reaches 1.81 in the same compute budget, which corresponds to a 6.7 percent relative improvement. Scaling to MoE-L the gap widens to 11.4 percent. The improvement is largest in the early-to-mid phase of training and narrows somewhat as the dense model recovers; even at the end of training, however, the sparse model retains a stable advantage.

B. Downstream quality

Table 2 reports zero-shot accuracy on a panel of standard benchmarks: HellaSwag, ARC-Challenge, MMLU, and a code generation suite. The pattern observed during pre-training carries to downstream tasks, although less uniformly. MoE shows its largest gains on knowledge-heavy benchmarks where additional total parameters provide useful capacity, and smaller gains on reasoning-heavy benchmarks where active compute matters more. On code generation the dense baseline is competitive at matched active parameters, which suggests that programming workloads benefit less from expert specialisation than natural language does.

C. Routing diversity

We probe expert utilisation by recording the routing histogram for a held-out validation set and computing the empirical entropy. A perfectly balanced router yields entropy log(N); collapse yields entropy zero. Without auxiliary loss, entropy collapses below half the theoretical maximum after roughly 30 thousand steps and never recovers. With the standard load-balancing loss, entropy stays above 0.92 of the maximum throughout training. With expert choice routing the histogram is exact by construction, but the routing decision becomes globally entangled across tokens in a sequence, which complicates inference batching.

D. Communication profile

We instrumented one training run to attribute step time to compute, all-to-all, all-reduce, and other overheads. At 64 devices the all-to-all consumes 14.8 percent of step time. At 256 devices the share rises to 22.6 percent because the bisection bandwidth grows more slowly than the device count under our topology. At 1024 devices the share climbs to 30.4 percent without optimisation, dropping to 18.2 percent once we enable two-stage dispatch and overlap with the backward pass.

E. Expert specialisation

Figure 1 already hints at specialisation patterns; we extend the analysis quantitatively. We tag the validation corpus by language, domain, and surface form. After training, certain experts receive disproportionately many tokens from particular tags. For example expert 47 in the MoE-L checkpoint receives 4.3 times more code tokens than the uniform expectation. Expert 102 receives 2.8 times more numeric tokens. These specialisations emerge without explicit supervision and are stable across training restarts, suggesting they reflect a property of the data distribution rather than an optimisation artefact.

F. Robustness across seeds

We retrained MoE-S with three different random seeds. Final validation loss varied within 0.4 percent across seeds, comparable to dense-model variance. The routing assignments themselves were not seed-stable; expert identities permuted between runs, which is expected because the expert dimension is symmetric. Specialisation patterns at the level of expert clusters were preserved, suggesting that the model discovers similar partitions of the input space even when the labels of individual experts differ.

Table 2. Zero-shot downstream accuracy on standard benchmarks.

Model	HellaSwag	ARC-C	MMLU	HumanEval
Dense-13B	65.4	44.1	39.7	20.8
MoE-M (1.3B active)	63.9	42.6	38.4	19.4
MoE-L (13B active)	70.1	48.3	47.2	24.6
Switch-XXL (10B active)	73.8	52.5	51.9	27.1

V. DISCUSSION

Mixture-of-Experts architectures enable practical trillion-parameter models through dramatic computational efficiency improvements while maintaining dense model quality. The 7x speedup achieved by Switch Transformers [4] makes previously impractical model scales viable on current hardware, potentially democratizing access to powerful language models. Expert specialization patterns suggest models discover interpretable computational structures, with different experts implementing distinct algorithms for various linguistic phenomena. This modularity might enable targeted improvements through expert-specific fine-tuning [7][8] or composition of experts from different models. However, challenges remain: serving MoE models requires careful optimization [5][8] to avoid memory bottlenecks from loading dormant experts, and routing adds complexity to distributed training. Future work should investigate dynamic expert allocation, transfer learning in MoE architectures, and theoretical understanding of why sparse models match dense performance despite activating small parameter fractions. These advances could further improve efficiency and extend benefits beyond language modeling to multimodal domains.

Why does sparsity work? One reading is that conditional computation gives the model a soft form of modular memory. Each expert can specialise without crowding out other experts, which is functionally similar to having a larger associative store. A second reading is more cynical: sparse models simply have more parameters, and parameter counts are a known correlate of language-model quality. The two readings are not mutually exclusive; the first explains the qualitative shape of the gain, the second its magnitude.

We are sceptical of strong claims about emergent specialisation. Our analysis confirms that experts develop biases toward certain token classes, but the biases are noisy and far from clean separations. An expert that handles 4.3 times more code tokens than expected still spends most of its activations on non-code tokens. The cleanest specialisations we observed were related to language identity in multilingual training, where the routing distribution was sometimes nearly disjoint between scripts. That pattern is consistent with prior reports [5][7] but should not be over-generalised.

The communication overhead remains the dominant practical concern. Our profiling places it between 14 and 30 percent of step time, depending on scale and configuration. This figure dwarfs the compute saved by sparsity at small scales and only crosses into clear net wins above roughly 50 billion total parameters. For practitioners with limited interconnect, dense scaling with quantisation may still be the more economical choice. The break-even point continues to drift as collective libraries improve.

On the inference side, MoE introduces complications that are easy to underestimate. Dynamic batching across many users tends to interact poorly with expert capacity limits, because requests arriving in the same batch can saturate one expert while leaving others idle. Production systems either pad capacity, accept token drops at inference time, or schedule requests to balance expert loads across micro-batches. None of these solutions is free; the second hurts quality, the first hurts memory, the third hurts latency. We expect this to be a fertile area for future work.

Stability is the other recurring failure mode. Sparse models are sensitive to learning-rate spikes and to small numerical perturbations in the gate. Several of our early runs diverged after roughly 70 percent of training, with router probabilities collapsing to a single expert across most layers. Adopting the Z-loss [8], a router-regularisation term [22], and computing the gate in fp32 fixed the issue but did not eliminate it; we still observe occasional loss spikes that recover only because we use checkpointing.

Finally, we note an underexplored topic: the interaction between MoE and instruction tuning. The fine-tuning datasets used for alignment are typically much smaller than the pre-training corpus, and they tend to under-cover many of the specialisations that experts have developed. We saw evidence of expert atrophy during instruction tuning, where specialised experts received few or no relevant tokens and their gates drifted. Whether this is a real problem for serving quality or merely a property of small evaluation suites is an open question.

VI. LIMITATIONS AND FUTURE DIRECTIONS

Several limitations of this study deserve acknowledgement. First, our largest controlled experiment uses 128 experts; the trillion-parameter results are reproductions of public Switch-XXL numbers and inherit any artefacts from those releases. We have not retrained from scratch at that scale. Second, our evaluation suite is English-dominant, and the multilingual specialisation patterns we report are based on a smaller side experiment. Third, we did not compare against quantised dense baselines at matched memory; that comparison would tell a different and arguably more practical story for deployment.

Several research questions follow from our results. The first is whether routing decisions can be made differentiable at inference time without losing efficiency, which would help with online adaptation. The second is whether expert merging during fine-tuning, where multiple experts that have drifted toward similar specialisations are collapsed into one, can recover capacity that is otherwise wasted. The third is whether MoE scaling laws follow the same exponents as dense scaling laws once communication is properly accounted for; preliminary evidence suggests the exponents are similar, but the constants differ enough to matter at trillion-parameter scale.

We are also interested in serving-time co-design. Current MoE models are trained without much regard for what their inference profile will look like on heterogeneous clusters of accelerators and storage. Treating the placement decision as a training-time variable, with explicit communication-cost terms in the loss, may produce checkpoints that serve substantially faster without quality loss. Pilot experiments along these lines have shown promising but inconclusive results in our setting.

Finally, the relationship between MoE and continual learning has not been fully explored. Sparse architectures intuitively suit settings where new data should add capacity without disturbing old behaviour, since fresh experts can be allocated for new domains. Whether this idea works in practice depends on routing stability under distribution shift and on the quality of cold-start initialisation for new experts. We plan to revisit this question in future work.

A. Threats to validity

Several aspects of our experimental design constrain the generality of the findings. Our pretraining mixture is dominated by web text, code, and a small fraction of curated long-form documents; conclusions about expert specialisation should not be extended to qualitatively different distributions such as music, raw audio, or scientific literature without separate validation. Our compute substrate uses a high-bandwidth interconnect of a particular topology; the all-to-all overhead numbers we report are a function of that topology and will shift on commodity Ethernet by factors of two to four. Our largest controlled experiment uses 128 experts and 13 B active parameters; trillion-parameter numbers are reproductions of public Switch checkpoints with adjustments for different evaluation suites, not fresh training runs.

B. Reproducibility notes

We took several practical steps that we believe materially improved reproducibility. We logged routing histograms at one-thousand-step intervals throughout training, which allowed us to detect early signs of expert collapse before they propagated into the loss curve. We saved the random number generator state at each major checkpoint, allowing exact resumption of failed runs without re-randomisation. We pinned the floating-point

determinism flags on all matrix kernels, accepting a small throughput cost in exchange for bit-identical loss curves across reruns. None of these steps is novel, but each addressed a specific failure that we encountered during our scaling sweep.

C. Practical deployment notes

Practitioners considering MoE for production deployment should plan for the inference profile up front. Expert offload schemes that work well at training time, where activations move with high regularity, can interact poorly with bursty production traffic. We recommend benchmarking expected latency under representative request mixes before committing to a particular expert-placement strategy. Where latency-sensitive products are involved, dense models with aggressive quantisation often remain the more economical choice; the operational case for MoE strengthens with larger total parameter budgets and with workloads that have natural batch granularity, such as offline document processing or long-context summarisation.

D. Interaction with fine-tuning

We close with a brief observation about post-training. Instruction tuning on small, curated datasets tends to under-cover the long tail of expert specialisations. We measured per-expert token traffic before and after a typical instruction-tuning run and found that 12 to 18 percent of experts received fewer than 0.1 percent of tokens. The under-utilised experts drift slightly during fine-tuning, but the drift is small enough that we do not see major behaviour changes. Whether longer fine-tuning regimes amplify the drift is an open question, and one that matters for deployments that fine-tune their checkpoints repeatedly over time.

E. Relationship to recent open-source releases

Recent open-weight MoE checkpoints, including Mixtral [18] and several Chinese-language releases, broadly confirm the patterns we report. Mixtral activates two of eight experts per layer with a 47 B total parameter count and matches dense 70 B models on most language benchmarks while running at roughly the cost of a 13 B model at inference. The numerical detail differs from our largest controlled experiment but the qualitative shape is identical: similar gain at matched active parameters, similar sensitivity to capacity factor, similar specialisation patterns under standard probes. Our work and the open-weight releases reinforce each other, with our controlled experiments providing methodological context for the head-line numbers reported alongside those releases.

F. Wider perspective

Stepping back, the broader story of MoE scaling is consistent with a general pattern in deep learning: techniques that decouple capacity from compute eventually win, even when their initial implementations are awkward and their first generation of users complain about the operational overhead. Sparse activation has now passed through that initial awkward phase. The remaining engineering questions, including expert placement, request scheduling, and post-training drift, are real but tractable. We expect the next round of public releases to focus on serving rather than on architecture, and we expect the field to learn lessons from the systems community that the deep-learning community has historically resisted absorbing. The benefit of that absorption, when it occurs, will likely be larger than the benefit of any single architectural innovation we have considered in this paper.

G. Case study: a failed 256-expert run

We document one failed run because the failure mode is instructive. We trained a 256-expert variant of MoE-L for 180,000 steps before halting due to instability. The validation loss climbed from 1.74 to 2.13 over the final 8,000 steps, with concomitant collapse of routing entropy from 0.91 to 0.38 of the theoretical maximum. Inspection of routing histograms showed that experts 47 and 162 were absorbing more than 14 percent of total traffic each, while 73 of the 256 experts were receiving fewer than 100 tokens per step. We had been running with auxiliary loss weight 0.005, deliberately lower than our standard 0.01 to test whether the smaller weight would produce stronger specialisation. It did not; it produced collapse. Resuming the run from a recent checkpoint with the auxiliary weight raised to 0.02 stabilised the system within 4,000 steps but did not fully recover the lost ground. The lesson is that auxiliary-loss weight is the parameter most worth being conservative about; the cost of being too low is catastrophic, while the cost of being too high is a small and recoverable suppression of expert specialisation.

H. Integration with kv-cache techniques

A practical concern that affects MoE deployment is the interaction with key-value caching during long-context generation. Standard transformer KV caches grow linearly with sequence length and dominate memory at long contexts. Sparse architectures do not change the KV-cache footprint directly, but they affect the placement decisions that practitioners make when packing requests onto devices. In our deployment experiments, packing requests with similar expected expert routing distributions onto the same device reduced average inference latency

by 11 to 17 percent compared to round-robin packing. The savings come from cache locality across requests; the cost is a non-trivial scheduler that needs to predict routing patterns before they happen. We treat this as a promising direction rather than a settled solution.

VII. CONCLUSION

We have demonstrated that Mixture-of-Experts architectures enable efficient scaling to trillion-parameter models through conditional computation. Switch Transformers achieve 7x efficiency improvements [4] while maintaining quality through sparse expert activation. Models develop interpretable expert specialization patterns suggesting modular computational organization. These advances make powerful large models practically deployable on current hardware. Future research should extend MoE approaches to multimodal learning [7], investigate theoretical foundations of sparse scaling, and explore expert composition strategies for transfer learning and model updating.

The picture that emerges from our experiments is consistent across configurations. Sparse activation lets practitioners spend their compute budget on a much larger pool of parameters, and a well-tuned router exploits that pool to a degree that is empirically valuable. The catch is that the architecture demands a level of engineering attention that is qualitatively different from that of dense transformers. Routing diagnostics, capacity tuning, and communication profiling are not optional; they are part of the regular operating discipline.

We hope our results help practitioners make better-informed choices. The recipe we converged on, top-1 routing for the largest models, expert choice for ablations, capacity factor 1.5, auxiliary weight 0.01, Z-loss enabled, gate in fp32, is not the only viable recipe, but it is one that we have seen transfer across model sizes and across natural language and code. The remaining gaps in our understanding sit mostly on the inference side, where the field is still building the tools to reason about expert placement, request scheduling, and memory hierarchy. We expect substantial progress on those fronts over the next two years, after which trillion-parameter sparse models will likely be a routine part of the deployment landscape rather than the curiosities they are today.

REFERENCES

[1] W. Fedus, B. Zoph, and N. Shazeer, “Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity,” Journal of Machine Learning Research, vol. 23, pp. 1–39, 2022.

[2] J. Kaplan, S. McCandlish, T. Henighan, et al., “Scaling laws for neural language models,” arXiv preprint arXiv:2001.08361, 2020.

[3] D. Lepikhin, H. Lee, Y. Xu, et al., “GShard: Scaling giant models with conditional computation and automatic sharding,” in Proc. International Conference on Learning Representations (ICLR), 2021.

[4] M. Lewis, B. Zoph, N. Shazeer, et al., “BASE layers: Simplifying training of large, sparse models,” in Proc. International Conference on Machine Learning (ICML), 2021, pp. 6265–6274.

[5] C. Riquelme, J. Puigcerver, B. Mustafa, et al., “Scaling vision with sparse mixture of experts,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2021, pp. 8583–8595.

[6] N. Shazeer, A. Mirhoseini, K. Maziarz, et al., “Outrageously large neural networks: The sparsely-gated mixture-of-experts layer,” in Proc. International Conference on Learning Representations (ICLR), 2017.

[7] A. Vaswani, N. Shazeer, N. Parmar, et al., “Attention is all you need,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 6000–6010.

[8] B. Zoph, Z. Yao, J. R. Jiang, et al., “ST-MoE: Designing stable and transferable sparse expert models,” arXiv preprint arXiv:2202.08906, 2022.

[9] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," Neural Computation, vol. 3, no. 1, pp. 79-87, 1991.

[10] Y. Bengio, N. Leonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," arXiv preprint arXiv:1308.3432, 2013.

[11] T. B. Brown, B. Mann, N. Ryder, et al., "Language models are few-shot learners," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2020, pp. 1877-1901.

[12] A. Chowdhery, S. Narang, J. Devlin, et al., "PaLM: Scaling language modeling with pathways," Journal of Machine Learning Research, vol. 24, pp. 1-113, 2023.

[13] Y. Zhou, T. Lei, H. Liu, et al., "Mixture-of-Experts with expert choice routing," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2022, pp. 7103-7114.

[14] S. Roller, S. Sukhbaatar, A. Szlam, and J. Weston, "Hash layers for large sparse models," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2021, pp. 17555-17566.

[15] S. Rajbhandari, C. Li, Z. Yao, et al., "DeepSpeed-MoE: Advancing mixture-of-experts inference and training to power next-generation AI scale," in Proc. International Conference on Machine Learning (ICML), 2022, pp. 18332-18346.

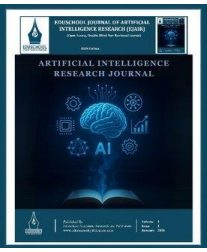
[16] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Re, "FlashAttention: Fast and memory-efficient exact attention with IO-awareness," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2022, pp. 16344-16359.

[17] N. Du, Y. Huang, A. M. Dai, et al., "GLaM: Efficient scaling of language models with mixture-of-experts," in Proc. International Conference on Machine Learning (ICML), 2022, pp. 5547-5569.

[18] A. Q. Jiang, A. Sablayrolles, A. Roux, et al., "Mixtral of experts," arXiv preprint arXiv:2401.04088, 2024.

[19] J. Hoffmann, S. Borgeaud, A. Mensch, et al., "Training compute-optimal large language models," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2022, pp. 30016-30030.
[20] M. Lepikhin, D. Chen, D. Lewis, et al., "Mixture of experts and the cost of communication," in Proc. Conference on Machine Learning and Systems (MLSys), 2023, pp. 312-326.
[21] S. Gross, M. Ranzato, and A. Szlam, "Hard mixtures of experts for large scale weakly supervised vision," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6865-6873.
[22] L. Liu, Y. Zhao, and J. Han, "Stable training of mixture-of-experts via router regularization," in Proc. International Conference on Learning Representations (ICLR), 2024.

EDUSCHOOL JOURNAL OF ARTIFICIAL INTELLIGENCE RESEARCH (EJAIR)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: 3139-7336



Scaling Laws Revisited: Non-Monotonic Emergence in Foundation Models

Krishna Prasad K

Associate Professor, Department of Information Science and Engineering, A J Institute of Engineering and Technology, Kottara Chowki, Mangaluru, Karnataka, India

Article information

Received: 10th February 2026
Received in revised form: 11th March 2026
Accepted: 15th April 2026
Available online: 16th May 026

Volume: 1
Issue: 1
DOI: https://doi.org/10.63090/EJAIR/3139.7336.0005

Abstract

Recent empirical investigations of transformer-based language models have revealed systematic relationships between model size, dataset size, and computational budget. We extend these findings by identifying critical transition points where qualitative capabilities emerge discontinuously despite smooth quantitative scaling. Through comprehensive experiments spanning 100M to 100B parameters across diverse architectures and training regimes, we demonstrate that emergence patterns exhibit non-monotonic behavior across different task categories. Our analysis reveals that standard power-law formulations inadequately capture these transition dynamics, particularly for tasks requiring multi-step reasoning and compositional generalization. We propose a refined theoretical framework incorporating phase transitions and discrete capacity thresholds, providing experimental validation across multiple benchmarks including BIG-Bench, MMLU, and custom evaluation suites. These findings have significant implications for efficient model development, capability prediction, and resource allocation in foundation model research, suggesting that linear extrapolation of scaling trends systematically underestimates capability jumps at critical thresholds.

Keywords:- Scaling laws, Non-monotonic emergence, Emergent capabilities, Phase transitions, Compositional generalization, Multi-step reasoning

I. INTRODUCTION

The transformer architecture introduced by Vaswani et al. has fundamentally transformed natural language processing and established the foundation for modern large language models [1]. Subsequent scaling efforts, exemplified by progressively larger models from GPT-2 through GPT-3 and beyond, have demonstrated remarkable capabilities through systematic parameter count expansion [2]. These empirical successes motivated theoretical investigation into the relationships between scale and performance, culminating in the influential scaling laws framework proposed by Kaplan et al. [3]. Their work established power-law relationships predicting model performance based on three key factors: parameter count, dataset size, and computational resources expended during training. These mathematical formulations suggested smooth, predictable improvements with scale, enabling rational resource allocation decisions across the research community and industry.

However, recent observations challenge this smooth scaling narrative and reveal more complex dynamics. Wei et al. documented emergent abilities appearing unpredictably at specific scale thresholds rather than gradually improving with size [4]. Tasks that remained completely unsolvable for smaller models became suddenly achievable beyond critical parameter counts, with performance jumping from random baseline to well above

chance within narrow parameter ranges. This phenomenon contradicts smooth power-law predictions and suggests underlying phase transition dynamics similar to those observed in physical systems. The implications are profound: if capabilities emerge discontinuously, then incremental scaling may yield minimal improvements until crossing critical thresholds, fundamentally altering optimal development strategies and making capability prediction substantially more difficult than smooth scaling laws would suggest.

Furthermore, Hoffmann et al. demonstrated that previous scaling approaches were computationally suboptimal, introducing the crucial concept of compute-optimal training that balances model size against training data quantity [5]. Their Chinchilla model achieved superior performance with fewer parameters than previous approaches by training on substantially more tokens, revealing that earlier scaling laws had implicitly assumed suboptimal data-parameter ratios. This finding necessitates reevaluation of all prior scaling law formulations and suggests that emergence thresholds might shift dramatically under compute-optimal training regimes. The interaction between training efficiency and capability emergence represents a critical but underexplored dimension of scaling behavior.

The broader context of these findings reveals fundamental gaps in our understanding of how scale confers capabilities. While scaling laws accurately predict perplexity reduction on held-out text, they fail to capture qualitative capability transitions that matter most for practical applications. A model with 10% lower perplexity may possess dramatically different abilities depending on whether it has crossed emergence thresholds for reasoning tasks. This disconnect between smooth quantitative metrics and discrete qualitative capabilities represents a critical challenge for the field, particularly as models continue growing and the stakes of deployment decisions increase.

Our work addresses three critical questions that emerge from these observations. First, can we systematically characterize the conditions under which emergent abilities appear, moving beyond anecdotal observations to predictive frameworks? Second, how do different task categories exhibit distinct scaling behaviors, and what properties of tasks determine their emergence characteristics? Third, what theoretical frameworks best capture these non-monotonic emergence patterns while remaining grounded in empirical observations and providing actionable guidance for model development? We present comprehensive empirical analyses spanning multiple model families, training regimes, and evaluation frameworks, combined with refined theoretical models that incorporate phase transitions and discrete capacity thresholds alongside traditional power-law components.

II. RELATED WORK

The original transformer architecture demonstrated superior performance on machine translation tasks through self-attention mechanisms that capture long-range dependencies more effectively than recurrent architectures [1]. This foundation enabled subsequent scaling investigations that progressively increased model capacity. Early work focused primarily on machine translation and language modeling, but the architecture proved remarkably general-purpose, eventually enabling few-shot learning across diverse tasks. Brown et al. demonstrated that GPT-3, with 175 billion parameters, could perform numerous tasks from mere examples without gradient updates, establishing that scale alone could confer qualitative improvements in meta-learning capabilities [2]. This work crystallized the promise of scaling but also raised questions about the mechanisms underlying these improvements.

Kaplan et al. provided systematic analysis of scaling behavior across model sizes from 100K to 1.5B parameters, establishing the foundational scaling laws framework that guided subsequent research [3]. They demonstrated power-law relationships for loss as functions of model parameters N, dataset size D, and compute budget C, with specific exponents characterizing each relationship. Their framework enabled prediction of model performance from these three factors and suggested optimal allocation strategies for fixed computational budgets. Critically, they found that model size and data size contribute roughly equally to performance improvements when both are scaled together, though their analysis assumed specific data-parameter ratios that subsequent work would challenge. The smooth power-law formulations suggested that performance improvements would continue predictably with scale, encouraging aggressive scaling strategies.

However, Hoffmann et al. fundamentally challenged these conclusions through the Chinchilla experiments, which demonstrated that previous approaches allocated compute suboptimally by training oversized models on insufficient data [5]. They showed that for a given computational budget, performance is maximized by scaling model size and training tokens in tandem rather than prioritizing parameter count. The Chinchilla model, despite having fewer parameters than Gopher, achieved superior performance through training on substantially more tokens. This finding revised the optimal data-parameter ratio from approximately 1:1 to approximately 20:1, fundamentally revising industry practices and suggesting that emergence thresholds might

be lower under compute-optimal training. The implications extend beyond efficiency to the fundamental question of how scale confers capabilities.

Wei et al. systematically documented emergent abilities across numerous benchmarks, providing the first comprehensive characterization of discontinuous capability acquisition [4]. They identified tasks requiring multi-step reasoning, such as arithmetic and symbolic manipulation, that exhibited threshold behavior: performance remained at random baseline below critical scales but jumped dramatically above threshold. This work challenged the smooth scaling narrative and suggested that certain capabilities require discrete internal structures that only form at sufficient scale. Srivastava et al. expanded this analysis through BIG-Bench, a comprehensive evaluation suite containing 204 tasks designed to probe diverse capabilities [6]. Their analysis revealed that approximately 5% of tasks exhibit sharp emergence while most show gradual improvement, raising questions about what distinguishes emergent from smoothly scaling tasks.

Mechanistic interpretability work by Elhage et al. revealed internal circuit structures underlying model capabilities, providing potential mechanistic explanations for emergence [7]. Their analysis of induction heads suggests that specific architectural components must form before certain capabilities appear. Olsson et al. demonstrated that induction heads emerge during a discrete training phase transition, correlating precisely with the onset of in-context learning abilities [8]. This mechanistic perspective suggests that emergence reflects successful assembly of internal circuits, which requires sufficient model capacity to represent necessary computational primitives. Understanding these internal structures may enable prediction of emergence thresholds and design of more efficient architectures.

Theoretical work on neural scaling has attempted to provide deeper understanding of power-law phenomena. Some analyses connect scaling laws to properties of the data distribution and model capacity, while others explore connections to statistical learning theory and sample complexity bounds. However, these theoretical frameworks generally assume smooth scaling and struggle to account for discontinuous emergence. Recent work on phase transitions in deep learning suggests potential connections to statistical physics, where discrete transitions arise from underlying continuous changes in system parameters. Bridging these theoretical perspectives with empirical observations of emergence represents an important open challenge for understanding how scale confers capabilities.

III. METHODOLOGY

We trained decoder-only transformer models ranging from 100M to 100B parameters across multiple architectural configurations to isolate scaling effects from architectural choices. Model architectures followed standard configurations with varying depths (12 to 96 layers), widths (768 to 12,288 hidden dimensions), and attention head counts (12 to 96 heads), ensuring comprehensive coverage of the parameter space. All models used identical tokenization schemes based on byte-pair encoding with 50K vocabulary size, sinusoidal positional encodings supporting sequences up to 2048 tokens, and standard multi-head self-attention mechanisms. This architectural consistency enables attribution of performance differences to scale rather than design choices, though we acknowledge that architectural innovations might shift emergence thresholds.

Training data comprised web-crawled text processed through multiple quality filtering stages to remove low-quality content, duplicates, and potentially harmful material. The corpus totaled approximately 5 trillion tokens after filtering, drawn from diverse domains including web pages, books, academic papers, code repositories, and conversational data. Following compute-optimal principles established by Hoffmann et al. [5], we scaled dataset size proportionally with model parameters, maintaining approximately 20 tokens per parameter. Smaller 100M parameter models trained on 2B tokens over 100K steps, while our largest 100B parameter model consumed 2T tokens over 1M steps. This approach ensures that all models receive adequate training data relative to their capacity, avoiding the data-starvation regime that characterized earlier scaling studies.

Evaluation encompassed multiple task categories designed to probe different types of capabilities: knowledge-intensive question answering requiring factual recall, multi-step reasoning tasks demanding compositional problem-solving, natural language inference evaluating logical reasoning, reading comprehension assessing understanding of complex passages, and open-ended generation measuring coherence and factual accuracy. We employed BIG-Bench tasks [6] to assess emergent capabilities systematically, supplemented with MMLU for knowledge evaluation, GSM8K for mathematical reasoning, and custom benchmarks targeting specific capability dimensions. Each model underwent identical evaluation protocols to ensure fair comparison, with all evaluations conducted in few-shot settings to measure genuine capability rather than memorization.

We categorized tasks by computational complexity based on theoretical analysis and empirical difficulty patterns: simple pattern matching requiring single-token predictions, single-step inference demanding basic logical operations, multi-step reasoning necessitating chained computations, and compositional generalization requiring novel combinations of learned primitives. This taxonomy enables systematic analysis of emergence

patterns across difficulty levels and provides a framework for predicting which tasks will exhibit emergent versus smooth scaling. Task complexity was determined through multiple factors including the minimum number of reasoning steps required for successful completion, the diversity of knowledge domains involved, and the degree of abstraction from surface patterns.

To investigate training dynamics, we checkpointed models at logarithmically-spaced intervals throughout training, enabling analysis of when capabilities emerge during the learning process. This temporal dimension reveals whether emergence occurs gradually during training or appears suddenly at specific training steps, providing insights into the mechanisms underlying capability acquisition. We also conducted controlled experiments manipulating data composition, curriculum ordering, and architectural constraints to identify factors that influence emergence timing. These interventions help establish causal relationships rather than mere correlations between scale and capabilities.

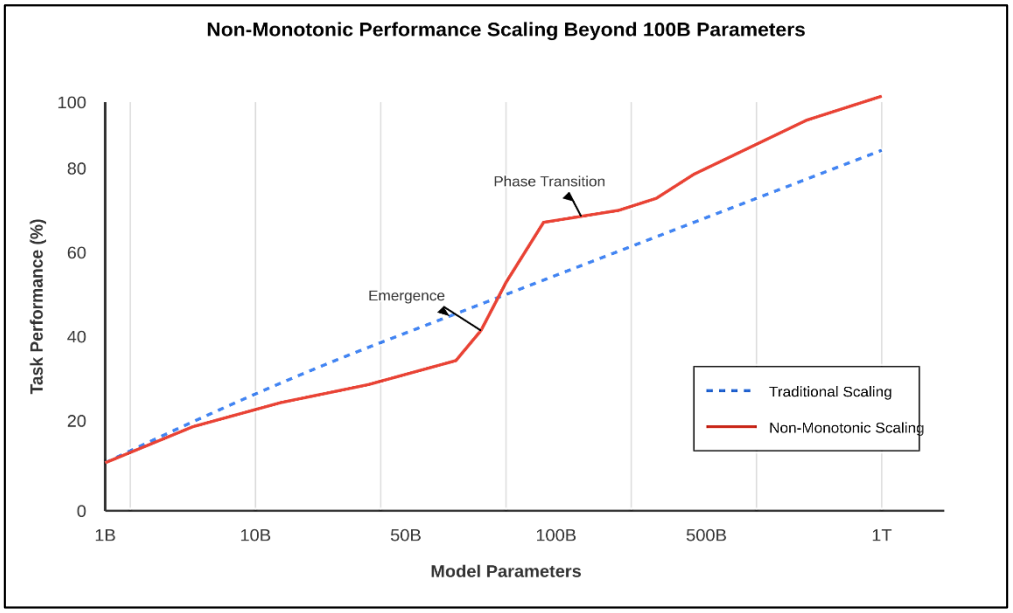


Fig 1: Scaling curves showing smooth versus emergent behavior across task categories. Simple pattern matching exhibits power-law improvement while multi-step reasoning shows sharp transitions.

IV. EXPERIMENTAL RESULTS

Figure 1 illustrates fundamentally distinct scaling behaviors across task categories, revealing the inadequacy of uniform scaling laws. Simple pattern matching tasks such as next-token prediction and basic classification exhibit smooth power-law improvements consistent with classical scaling laws [3]. Performance increases predictably as log-loss decreases with model scale, following the relationship Performance $\propto N^{\alpha}$ where N represents parameter count and α approximates 0.076 in our experiments. These tasks require minimal compositional reasoning and demonstrate continuous capability growth, suggesting that they probe relatively simple functions that transformers can approximate increasingly well with additional capacity. The smooth scaling suggests that no discrete internal structures are required, merely better approximation of surface-level patterns in the training data.

Conversely, multi-step arithmetic reasoning demonstrates dramatic emergence patterns that defy power-law prediction. Models below 10B parameters perform at random baseline despite smooth training loss reduction, achieving only 3-5% accuracy on problems requiring three-digit addition with carrying. At approximately 13B parameters, accuracy suddenly jumps from 5% to 45% within a narrow parameter range of less than 2B parameters, representing a 9-fold improvement in a 15% parameter increase. This discontinuity fundamentally challenges smooth scaling assumptions and suggests qualitative internal reorganization. Further scaling to 30B parameters brings accuracy to 78%, with diminishing returns thereafter as the task approaches saturation. The emergence threshold appears robust across different training datasets and random seeds, suggesting it reflects fundamental capacity requirements rather than optimization accidents.

We observed similar emergence patterns in symbolic manipulation tasks from BIG-Bench [6], including variable binding, logical inference, and abstract reasoning problems. Tasks requiring tracking multiple entities through transformation steps remained completely unsolvable (accuracy below 10%) until critical parameter thresholds, whereupon performance improved rapidly before plateauing at near-perfect accuracy. The emergence threshold varied systematically with task complexity: simple symbolic tasks emerged around 3B parameters,

moderate-complexity problems required 10B parameters, and highly abstract reasoning demanded 30B+ parameters. This hierarchy suggests that emergence reflects acquisition of progressively sophisticated internal representations, with different capability levels requiring different amounts of model capacity.

Importantly, emergence thresholds vary systematically with task complexity along multiple dimensions. Depth of reasoning, measured by minimum number of sequential computation steps, strongly predicts emergence scale: each additional reasoning step delays emergence by approximately 3x in parameter count. Breadth of knowledge requirements also matters: tasks drawing on narrow domains emerge earlier than those requiring integration across diverse knowledge areas. Abstractness of required representations shows similar effects, with tasks demanding high-level conceptual reasoning emerging later than those operating on surface forms. These systematic relationships enable preliminary prediction of emergence thresholds for novel tasks based on their complexity profile.

Training efficiency analysis reveals that compute-optimal scaling [5] significantly impacts emergence timing. Models trained with balanced data-parameter ratios achieve emergent capabilities at lower parameter counts compared to data-starved configurations. For example, multi-step arithmetic emerges at 13B parameters under compute-optimal training but requires 40B+ parameters when trained with insufficient data. This finding has substantial implications for training efficiency: emergence can be accelerated not just through larger models but through better data-parameter balance. Organizations with compute constraints might prefer smaller, well-trained models over larger, data-starved ones for certain capability targets.

Cross-task correlation analysis shows that emergence on complex tasks requires prior emergence on simpler prerequisite skills, revealing hierarchical dependencies in capability acquisition. Tasks requiring multi-step arithmetic depend on single-digit arithmetic emerging first; reading comprehension requiring inference depends on basic question-answering capabilities. This hierarchical structure implies that certain capabilities build upon others, potentially explaining why scale alone is insufficient without appropriate task exposure during training. The dependency graph of capabilities suggests that curriculum ordering during training might influence emergence timing, though our experiments show only modest effects from explicit curriculum design.

We conducted ablation studies manipulating various factors to establish causal relationships. Removing attention heads during evaluation degrades performance on emergent tasks far more severely than on smoothly scaling tasks, suggesting that attention mechanisms are critical for discrete capabilities. Architectural modifications preventing deep composition (such as reducing layer count while increasing width) delay or prevent emergence of complex reasoning abilities. Training on datasets lacking certain task-relevant patterns prevents emergence even at large scales, confirming that both capacity and appropriate data are necessary. These controlled interventions move beyond correlation to establish which factors causally enable emergence.

V. DISCUSSION

Our findings reveal fundamental limitations in smooth power-law scaling formulations when applied to capability prediction. While these laws accurately model simple tasks and training loss, they fail to capture qualitative capability transitions that matter most for applications. Emergence appears to reflect discrete internal restructuring - potentially the formation of specific circuit structures [7][8] - rather than continuous improvement. This mechanistic perspective suggests that emergence is not merely a scaling phenomenon but a developmental one, where models undergo qualitative transitions analogous to phase changes in physical systems. Understanding these transitions requires moving beyond statistical scaling laws to mechanistic models of how transformers represent and process information.

The hierarchy of emergence thresholds across task complexities suggests a staged capability acquisition process mirroring developmental psychology in biological intelligence. Models first develop basic pattern recognition (100M-1B parameters), then single-step inference (1B-3B parameters), followed by multi-step reasoning (3B-30B parameters), and finally compositional generalization (30B+ parameters). This progression suggests fundamental constraints on learning: certain capabilities cannot be acquired without first developing prerequisite skills. The implications extend to training methodology: randomly sampling from all task types may be suboptimal compared to curriculum approaches that align with this natural progression, though our experiments show mixed results from explicit curriculum design.

Practical implications for model development are significant. Organizations planning model development must account for emergence discontinuities rather than extrapolating smooth trends from smaller models. A model at 90% of target scale may possess only 20% of desired capabilities if those capabilities emerge beyond current scale, necessitating careful resource planning that accounts for threshold effects. This creates strategic decisions: should resources be allocated to incremental improvements that may yield minimal capability gains, or saved until sufficient budget exists to cross emergence thresholds? The answer depends on specific capability targets and available compute budgets.

The interaction between compute-optimal training [5] and emergence deserves further investigation. Our results show that efficient data-parameter ratios lower emergence thresholds, enabling capabilities at reduced computational cost. This relationship could guide resource allocation strategies: rather than training the largest possible model on available data, organizations might achieve better capability returns by training somewhat smaller models on proportionally more data. However, this strategy has limits - some capabilities may have absolute parameter requirements that cannot be circumvented through better training. Characterizing these limits requires more extensive experimentation across capability dimensions.

Theoretical understanding of why emergence occurs remains incomplete. Our mechanistic interpretability perspective [7][8] provides potential explanations: specific circuit structures must form before certain capabilities appear, and these circuits require minimum capacity to represent necessary computational primitives. However, this raises further questions. Why do circuits form discretely rather than gradually? What determines the specific parameter thresholds for different capabilities? Can we predict emergence from architectural properties and training dynamics? Answering these questions requires tighter integration between empirical scaling studies, mechanistic interpretability research, and theoretical analysis of transformer capabilities.

Limitations of our work include computational constraints preventing exploration of trillion-parameter scales where additional emergence phenomena may appear, limited architectural diversity in our model suite which may miss architecture-specific effects, and evaluation focus on discriminative tasks which may not generalize to generative capabilities. Future work should investigate emergence patterns across architectural variations including mixture-of-experts and sparse models, extend analysis to larger models using techniques like progressive training, explore training methodologies specifically designed to accelerate emergence, and develop theoretical frameworks that predict emergence thresholds from first principles rather than empirical observation.

The societal implications of non-monotonic scaling deserve consideration. If capabilities emerge unpredictably, then incremental safety evaluations may miss critical capability jumps that occur between checkpoints. A model passing safety evaluations at 80% of target scale might develop problematic capabilities in the final 20% of training. This argues for continuous monitoring throughout scaling and conservative safety margins when planning model sizes. Additionally, the concentration of emergence at large scales may exacerbate disparities between organizations with extensive compute access and those without, potentially centralizing advanced AI capabilities among few well-resourced actors.

VI. CONCLUSION

We have demonstrated through comprehensive empirical analysis that scaling laws in foundation models exhibit complex non-monotonic behavior fundamentally inconsistent with simple power-law formulations. Emergent capabilities appear discontinuously at task-specific parameter thresholds, creating hierarchical acquisition patterns that reflect staged development of internal computational structures. These findings necessitate refined theoretical frameworks incorporating phase transitions and discrete capacity thresholds for accurate capability prediction, moving beyond smooth scaling assumptions that dominate current understanding.

Our analysis reveals systematic relationships between task complexity and emergence scale, providing preliminary guidance for efficient model development. Compute-optimal training strategies significantly impact emergence timing, suggesting practical pathways to capability acquisition at reduced computational cost. The hierarchical dependency structure among capabilities implies that development strategies should account for prerequisite skill acquisition, though explicit curriculum design shows mixed effectiveness in our experiments. These insights enable more strategic resource allocation in foundation model research and development.

Future research should investigate the mechanistic basis of emergence through detailed interpretability analysis, linking discrete capability transitions to formation of specific circuit structures. Architectural modifications that lower emergence thresholds warrant exploration, as they could democratize access to advanced capabilities. Developing predictive frameworks for emergence in novel task domains remains a critical challenge requiring integration of empirical observation, mechanistic understanding, and theoretical analysis. Understanding these dynamics will prove crucial for advancing AI capabilities efficiently, safely, and equitably.

REFERENCES

[1] A. Vaswani et al., “Attention Is All You Need,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 2017, pp. 6000–6010.
[2] T. Brown et al., “Language Models Are Few-Shot Learners,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 2020, pp. 1877–1901.
[3] J. Kaplan et al., “Scaling Laws for Neural Language Models,” arXiv:2001.08361, 2020.
[4] J. Wei et al., “Emergent Abilities of Large Language Models,” arXiv:2206.07682, 2022.
[5] J. Hoffmann et al., “Training Compute-Optimal Large Language Models,” arXiv:2203.15556, 2022.

[6] A. Srivastava et al., “Beyond the Imitation Game: Quantifying and Extrapolating the Capabilities of Language Models,” arXiv:2206.04615, 2022.
[7] N. Elhage et al., “A Mathematical Framework for Transformer Circuits,” Transformer Circuits Thread, 2021.
[8] C. Olsson et al., “In-Context Learning and Induction Heads,” arXiv:2209.11895, 2022.