The Decade of Attention

How Transformers Rebuilt Machine Learning, 2017–2027

Dr. Priya Iyengar

A Kelford Press Original

First published in 2026 by Kelford Press

ISBN 978-1-7398-2240-4 (Digital) ISBN 978-1-7398-2241-1 (Print) ISBN 978-1-7398-2242-8 (Audio)

kelfordpress.com

For the researchers and engineers who, between 2017 and 2027, did the unglamorous work of figuring out what attention actually was — and what to do with it.

Contents

Introduction

This is a book about a single architectural idea — the Transformer — and what happened in the decade after it was published. It is not a comprehensive textbook on machine learning. It is a narrative history of an idea: where it came from, what made it spread so quickly, where it stalled, what it changed, and where it appears to be heading.

The arc is short and remarkable. In June 2017, eight researchers at Google Brain and Google Research published a paper called "Attention Is All You Need." It introduced an architecture for sequence modelling that abandoned recurrence entirely. Within five years, almost every state-of-the-art model in natural language processing, and a growing number in vision and audio, was a Transformer of some kind. Within eight years, the

architecture had reshaped entire industries. Within ten, it had become the substrate for what people were beginning to call artificial intelligence in a more-than-marketing sense.

What follows is a working ML practitioner's guide to that decade — written for engineers and researchers who have used these systems professionally and want a clear account of how they got here.

Chapter 1: Before Attention — The Sequence Problem

To understand why "Attention Is All You Need" landed with the force that it did, you have to understand what it replaced. In 2014, the dominant architecture for sequence modelling was the recurrent neural network — specifically the Long Short-Term Memory cell introduced by Sepp Hochreiter and Jürgen Schmidhuber in 1997, and its slightly simpler cousin the Gated Recurrent Unit. RNNs processed sequences one token at a time, carrying a hidden state forward from position to position, in

principle capable of remembering arbitrarily long context but in practice constrained by the vanishing-gradient problem and the strictly sequential nature of their computation.

The state of the art for machine translation in 2014 was the sequence-to-sequence model — an encoder RNN that read the input sentence into a single fixed-size vector, and a decoder RNN that generated the output one token at a time conditioned on that vector. Ilya Sutskever, Oriol Vinyals, and Quoc Le's paper "Sequence to Sequence Learning with Neural Networks" demonstrated this approach for English-to-French translation and produced results comparable to the strongest statistical machine-translation systems of the time. It was a striking demonstration of the power of end-to-end neural learning. It was also obviously broken.

The break point was the bottleneck vector. Compressing an arbitrarily long input sentence into a fixed-dimensional representation meant that longer sentences were translated worse than shorter ones, in a way that did not happen for human translators or for the older statistical systems. Within a year, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio published "Neural Machine Translation by Jointly Learning to Align and

Translate" — the paper that introduced the attention mechanism. Instead of compressing the encoder's full output into one vector, the decoder at each step computed a weighted sum over all encoder hidden states, with the weights determined by a learned alignment function. The decoder could, in effect, look back at the input sentence and decide which words to pay attention to at each output step.

The Bahdanau attention mechanism solved the bottleneck problem and pushed machine-translation accuracy forward by a meaningful margin. It was bolted onto an RNN backbone, but the seed of the idea was there: attention, as a way of conditioning each output position on a learned distribution over the input. Three years later, Vaswani and his co-authors at Google would ask what would happen if attention was not bolted onto the RNN but replaced it entirely. The result is the architecture we have been working in ever since.

What is worth noting about this pre-history, before we move on, is that attention was not invented in 2017. It was invented in 2014, and the three years between Bahdanau's paper and the Transformer paper were filled with hybrid models — RNNs with attention, ConvNets with attention, every conceivable combination — none of which

suggested that attention alone might be sufficient. The 2017 paper's contribution was not the mechanism but the negation: the claim, demonstrated empirically, that recurrence was not needed at all.

Chapter 2: The Paper (June 2017)

The paper is called "Attention Is All You Need." Its authors are Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. It was published on arXiv on 12 June 2017 and presented at NeurIPS that December. It is eleven pages long. It introduces an architecture, demonstrates its superiority on two machine-translation benchmarks, and provides enough detail for a competent engineer to reimplement it from the paper alone — a property that is rarer in ML publications than it should be.

The architecture, the Transformer, is built from two primitives stacked in alternating layers: multi-head self-attention and position-wise feed-forward networks. Both operate on sequences of vectors, take sequences of the

same length as input, and produce sequences of the same length as output. Neither uses recurrence. Neither uses convolution. The entire model is permutation-equivariant by construction, which is why position has to be added back in via a separate positional encoding — the authors propose a sinusoidal one, though learned embeddings work nearly as well.

Multi-head attention is the mechanism the title refers to. At each layer, every position in the sequence emits three vectors — a query, a key, and a value — by linear projection. The attention output at each position is a weighted sum of the values, with the weights determined by the scaled dot product between that position's query and every position's key, normalised by softmax. The "multi-head" part runs this computation in parallel several times with different learned projections and concatenates the results. It is, mechanically, very simple. It is also, computationally, very different from what came before: every position can attend to every other position in a single layer, in parallel, with no temporal dependency.

The two consequences of this design dominated everything that followed. First, the entire sequence could be processed in parallel during training, on hardware

(GPUs and the emerging TPUs) that had been built for parallel arithmetic. RNNs were strictly sequential and underutilised modern accelerators by a large factor; Transformers saturated them. Second, the model's effective context window was determined by attention quality and memory, not by gradient propagation through time. Long-range dependencies were as cheap to model as short-range ones, in principle, because every position attended to every other position directly. In practice the quadratic cost of attention in sequence length put a real ceiling on this — a ceiling that the rest of this book is, in many ways, about negotiating.

The empirical contribution of the paper was a state-of-the-art result on English-to-German machine translation, surpassing the best published RNN-based systems while training in a fraction of the wall-clock time on the same hardware. This was the result that made people take notice. It was not the largest improvement in WMT history. It was the cheapest one to date. The cost-of-progress curve for sequence modelling had, with one paper, pivoted.

A historical note: the paper's title is a play on a Beatles lyric, and Łukasz Kaiser has said in interviews that the authors knew it was a striking title and chose it for that

reason. There is something fitting about the most consequential architecture paper of the decade being also the most quotable. The phrase "all you need" has been a fixture of ML paper titles ever since — usually ironically, sometimes not.

Chapter 3: The Pretraining Era — BERT, GPT-1, T5

The Transformer paper demonstrated the architecture on supervised sequence-to-sequence tasks. Within a year, two papers showed that the architecture's real superpower was not supervised learning but pretraining: training a very large model on a very large amount of unlabelled text, then fine-tuning the resulting representations for downstream tasks.

The first of these was OpenAI's "Improving Language Understanding by Generative Pre-Training," released in June 2018, in which Alec Radford and his collaborators trained a 117-million-parameter decoder-only Transformer on the BooksCorpus dataset using a next-

token-prediction objective. The resulting model — what would later be retroactively named GPT-1 — was then fine-tuned on a battery of GLUE benchmarks and improved the state of the art on most of them. The paper's framing was modest. The implication was not.

The more immediately consequential paper was Google's "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," published in October 2018 by Jacob Devlin and colleagues. BERT trained an encoder-only Transformer on Wikipedia and BooksCorpus using two objectives: a masked language-modelling task in which random tokens were replaced with a special MASK token and the model had to predict the originals, and a next-sentence-prediction task that turned out to be less important than the original paper suggested. BERT-Large, at 340 million parameters, was the largest pretrained NLP model in public release at the time. Within months of its publication it had displaced every prior approach on the GLUE leaderboard.

BERT's effect on practitioners was immediate and substantial. The standard playbook for a new NLP task became: download BERT-Large, swap the final classification head, fine-tune for a few epochs, deploy. Whole categories of feature engineering became obsolete

overnight. The "BERT line" — the year before and the year after BERT's release — is a real discontinuity in NLP benchmark progress, visible in any longitudinal plot of GLUE accuracy.

Behind the leaderboard sport, however, was a more important shift. The Transformer architecture turned out to be unusually well-suited to a particular learning regime: massive unsupervised pretraining on next-token or masked-token prediction, followed by smaller supervised fine-tuning. This combination was not specific to language. Within two years, Vision Transformers would be doing the same thing on ImageNet pretraining, and audio Transformers on raw waveform prediction, with similar results.

The third pillar of the pretraining era was Google's T5 — "Text-to-Text Transfer Transformer" — published in late 2019 by Colin Raffel and colleagues. T5's contribution was not architectural; it was conceptual. Every NLP task, the authors argued, could be reframed as a text-to-text problem: classification became "label this text," translation became "translate this text," summarisation became "summarise this text." T5 was an encoder-decoder Transformer trained on a massive cleaned-up version of Common Crawl called C4, then fine-tuned on

the unified text-to-text formulation across many tasks. The result was a model that achieved state of the art across the board with a single architecture and a single output format.

By the end of 2019, the basic shape of the next five years was visible. Encoder-only Transformers (BERT and descendants) for classification and retrieval. Decoder-only Transformers (GPT and descendants) for generation. Encoder-decoder Transformers (T5 and descendants) for tasks where input and output were both sequences. Pretrain at scale, fine-tune at task. The architecture was settled. What remained was scale.

Chapter 4: Scaling Up — GPT-2 and GPT-3

GPT-2, released by OpenAI in February 2019, was the first widely-discussed scaling result. The largest model in the family — 1.5 billion parameters — was nearly thirteen times the size of GPT-1, trained on a much larger and more diverse dataset (WebText, a corpus of outbound

Reddit links filtered for upvotes), and produced text of a quality that crossed an uncanny threshold for many readers. The model could continue a fictional passage, summarise a news article, or attempt a translation in zero-shot — all without task-specific fine-tuning. The capability was, by 2019 standards, qualitatively new.

OpenAI's initial decision not to release the full 1.5B-parameter model on the grounds of potential misuse drew sharp criticism from some quarters of the research community. The full weights were released later that year, after a staggered rollout. The episode is worth remembering not for the release-policy debate it sparked but for what it revealed about expectations: a 1.5-billion-parameter language model in 2019 was considered serious enough to merit a public conversation about release norms. The same model in 2024 would be a small, locally-runnable curiosity.

The real scaling result came in May 2020, with the GPT-3 paper. Tom B. Brown, Benjamin Mann, and a long list of co-authors at OpenAI scaled the architecture to 175 billion parameters and trained it on roughly 300 billion tokens drawn from Common Crawl, WebText2, Books1 and Books2, and Wikipedia. The paper's title — "Language Models are Few-Shot Learners" — described the central

empirical finding. GPT-3 could perform a wide range of NLP tasks reasonably well with only a few examples provided in its prompt, without any gradient updates. The phenomenon was called in-context learning. It changed the conversation about what these systems were.

In-context learning was not, in the GPT-3 paper, fully understood, and arguably is still not fully understood. The mechanism by which a frozen transformer can absorb a small number of input-output examples in its context and then generalise to new instances is the subject of an active research literature. What was clear empirically by 2020 was that this capability scaled smoothly with model size: small Transformers could not few-shot meaningfully, and large Transformers could. The threshold for usable few-shot performance varied by task. The progression from millions to billions of parameters, however, traced a recognisable curve.

GPT-3 was also the first public model whose capabilities visibly exceeded the imagination of the people who had trained it. The OpenAI authors, in the paper and in subsequent interviews, were repeatedly surprised by tasks the model could do that they had not anticipated. This pattern — emergence of capabilities at scale not specifically targeted by the training objective — would

become a defining feature of the next several years. It would also become the central source of unease about where the technology was headed.

What is sometimes lost in the GPT-3 retrospective is how genuinely expensive the model was to train. The compute budget, reported as several thousand petaflop-days, made the training run cost on the order of several million US dollars at then-prevailing cloud prices. For most of the field, the model was not just qualitatively new but financially inaccessible: only a handful of organisations could reproduce or extend the result. This created a new dynamic in the research community, in which capability and access concentrated rapidly into the hands of a few well-funded laboratories. The open-weight backlash that began in 2022 has roots in this concentration.

Chapter 5: The Compute Wall and Chinchilla's Lessons

By late 2020, the scaling pattern looked simple enough that researchers attempted to formalise it. The most influential of these efforts was OpenAI's "Scaling Laws for Neural Language Models" by Jared Kaplan and colleagues, which proposed that model loss followed a smooth power-law function of three variables — model size, dataset size, and compute — and that for a fixed compute budget there was an optimal allocation between the three. The Kaplan laws recommended scaling parameters roughly seven times faster than tokens. This recommendation was implemented in practice by training increasingly large models on relatively modest data.

Two years later, DeepMind's "Training Compute-Optimal Large Language Models" — the Chinchilla paper, by Jordan Hoffmann and colleagues — corrected the Kaplan analysis with a more careful experimental methodology and reached the opposite conclusion. For a fixed compute budget, the Chinchilla authors argued, you should scale parameters and training tokens at roughly equal rates. The implication was that most existing frontier models were significantly undertrained. The 70-billion-parameter Chinchilla model, trained on 1.4 trillion tokens, outperformed Gopher (280B parameters, 300B tokens) on nearly every benchmark.

The Chinchilla correction did several things simultaneously. It explained why some smaller models had been outperforming larger ones. It made the case for investing in larger, more diverse training corpora. It briefly redirected the entire industry's training methodology. And it set up the next several years of frontier-model development, in which the dominant lever for capability improvement was not parameter count but data quality and quantity.

The compute wall, more colloquially, was the observation that training compute could not grow forever at the rates seen between 2018 and 2022. The largest training runs of 2022 were consuming tens of millions of dollars of cluster time; the largest of 2024 would consume hundreds of millions; the largest projected for 2026–2027 would approach a billion. At each step, the population of organisations capable of running such a job shrank. By 2024, frontier-scale pretraining was effectively the province of five to ten organisations globally, all with substantial cloud-infrastructure backing or sovereign sponsorship.

This concentration had two effects worth flagging. First, it pushed an enormous amount of frontier research into a small number of corporate laboratories, with predictable

consequences for the rate of public publication. Second, it sharpened the focus of academic and open-source ML research onto problems that did not require frontier compute — inference efficiency, fine-tuning, evaluation methodology, mechanistic interpretability, and the long tail of architectural variants we discuss in Chapter 8.

The Chinchilla paper is, with hindsight, the inflection point at which the conversation shifted from "how much larger can we make these things" to "how much better can we make the data and the recipe." That shift was overdue.

Chapter 6: Alignment and ChatGPT

By late 2021, OpenAI had a model — InstructGPT — that took base GPT-3 and fine-tuned it using a procedure called reinforcement learning from human feedback, or RLHF, originally developed by Paul Christiano and colleagues several years earlier. The procedure was, in outline: collect a dataset of human preferences over model outputs, train a reward model to predict those preferences, then use reinforcement learning (specifically

Proximal Policy Optimisation) to fine-tune the language model to produce outputs that the reward model scored highly.

InstructGPT made GPT-3 substantially more useful for the kinds of instruction-following tasks that mattered to users — answering questions, writing summaries, refusing harmful requests — without changing the underlying capability of the model in any dramatic way. The technique was not introduced in the InstructGPT paper; what the InstructGPT paper showed was that it scaled, and that the resulting model was preferred by human raters over the much larger base GPT-3 across a wide range of tasks.

On 30 November 2022, OpenAI released ChatGPT, a conversational interface built around a successor to the InstructGPT model. The product page was minimal. The expectations inside OpenAI were, by multiple accounts, modest — a research preview that might attract some hundreds of thousands of users in its first month. Within five days the service had a million users. Within two months it had a hundred million. The release crossed a threshold that prior language-model releases had not: it was a product, not a paper, and it was good enough that ordinary people would use it on purpose.

The technical contribution of ChatGPT was, strictly speaking, modest. The architecture was the standard decoder-only Transformer. The pretraining was conventional next-token-prediction on a large corpus. The fine-tuning was RLHF in the manner of InstructGPT, with some additional curation. What was novel was the combination, plus a chat-style interface that turned every interaction into a structured dialogue. The cumulative effect was a system that felt categorically different from anything the public had used before.

The downstream effects of ChatGPT's release on the rest of the industry were profound. Google, which had been more cautious about deploying its Transformer-based chat systems publicly, accelerated work on what would become Bard and later Gemini. Anthropic's Claude entered public availability in early 2023. Meta and many smaller laboratories restructured their roadmaps around conversational chat assistants. Within eighteen months of ChatGPT's release, the dominant interaction model with frontier AI was a chat box. This was not architecturally inevitable; it was a product decision that reshaped the field.

The alignment work that made ChatGPT possible has also been the subject of significant subsequent research and refinement. Direct Preference Optimisation, introduced by Rafael Rafailov and colleagues in 2023, demonstrated that the reward-model + PPO pipeline could be simplified into a single supervised objective with comparable results. Constitutional AI, introduced by Anthropic in the same period, used model-generated critiques and revisions to reduce the human-labelling burden. Reinforcement Learning from AI Feedback (RLAIF) extended this further. By 2024, the post-training pipeline at every major lab had grown into a multi-stage affair of supervised fine-tuning, preference learning, and reinforcement-learning passes, often with each stage producing data for the next. The base pretrained model, the thing that the GPT-3 paper described, became one input into a much longer pipeline.

Chapter 7: The Open-Weights Awakening

For roughly five years — from the release of GPT-2 in 2019 to the early 2024 — frontier-scale language-model weights were not generally released to the public. The largest open-weight models of the period were Eleuther AI's GPT-Neo and GPT-J series, BigScience's BLOOM, and Meta's OPT-175B, each meaningful research efforts but lagging the closed frontier by a substantial margin in both capability and adoption.

This changed with the release of Llama in February 2023, and then more decisively with Llama 2 in July 2023. Meta's decision to release the weights of a 7B–70B-parameter Transformer trained on a frontier-comparable data regime — initially under a research-only licence, then under a commercially permissive one — opened the dam. Within months, the open-source community had produced a vast ecosystem of fine-tunes, quantised versions, inference frameworks, and applied derivatives. The Llama 2 release became, in retrospect, the founding moment of a real open-weight ecosystem at the frontier of capability.

Mistral AI followed in late 2023 with Mistral 7B and the Mixtral 8x7B Mixture-of-Experts model, both released under fully permissive Apache 2.0 licences and both punching well above their parameter weight on standard

benchmarks. Alibaba's Qwen series, DeepSeek, the smaller LLMs from labs around the world, and many community fine-tunes followed. By mid-2024, a working developer could deploy a strong language model entirely on their own hardware, with no API dependencies, using weights released by their original training organisation.

The open-weight movement had three structural effects on the field worth flagging here. First, it created a robust ecosystem of inference frameworks — vLLM, llama.cpp, MLC, TensorRT-LLM — that drove inference efficiency improvements at a much faster rate than would have been possible inside any single lab. The competition between these frameworks, all targeting the same set of public weights, produced order-of-magnitude improvements in inference cost between 2023 and 2025. Second, it democratised research on fine-tuning, alignment, and capability extension. The set of people who could meaningfully experiment with a 70B-parameter model grew from a few hundred at the major labs to tens of thousands in industry and academia. Third, it shifted the strategic landscape for frontier-model developers. Closed-weights advantage became, in many product categories, a thinner margin than its holders had hoped.

The line between "open weights" and "open source" deserves a brief note. The vast majority of so-called open-weight models in the 2023–2025 period released model weights and basic inference code but did not release training data, training code, or pretraining recipes. The DeepSeek and OLMo models were notable partial exceptions. This distinction matters: a developer can deploy and fine-tune an open-weight model, but cannot necessarily reproduce or audit its training. The conflation of "open weights" with "open source" in popular discourse obscured this throughout the period.

By 2026, the dominant pattern in industry deployment had become hybrid: frontier-model API calls for the hardest tasks, open-weight models for the long tail of routine and latency-sensitive cases, and the line between the two moving outward as open-weight quality improved. This is the working environment most ML practitioners inhabit today.

Chapter 8: Architectural Variants — MoE and the State-Space

Detour

For the first six years of the Transformer's life, architectural innovation was largely confined to scaling and incremental refinements: better normalisation (Pre-Norm replacing Post-Norm), better attention numerics (FlashAttention), better positional encodings (RoPE replacing sinusoidal), better activation functions (SwiGLU). The core block — multi-head self-attention plus position-wise feed-forward — remained essentially what Vaswani et al. had described in 2017.

The two meaningful architectural departures of the 2023–2025 period were Mixture-of-Experts and structured state-space models.

Mixture-of-Experts replaces the dense feed-forward sublayer in each Transformer block with a set of expert feed-forward sublayers and a learned router that, for each token, selects a small subset (typically the top one or two) of experts to route to. The result is a model with a much larger total parameter count than a dense equivalent, but with computational cost per token comparable to a much smaller dense model. The Switch Transformer paper from Google in early 2021 demonstrated the approach at scale; Mixtral 8x7B from Mistral in late 2023 was the first

widely-adopted open-weight MoE model; by 2024 most frontier-scale models were MoE in some form, though the major labs did not always confirm this in public.

The appeal of MoE is essentially economic. A 70-billion-active-parameter MoE model can be trained and served at roughly the cost of a 13-billion-parameter dense model while achieving capability closer to that of the much larger one. The trade-offs are non-trivial: MoE models have higher memory requirements (all experts must be in memory even though only a few are active per token), more complex distributed-training dynamics (the routing function must be load-balanced across experts), and more difficult fine-tuning behaviour. By 2025, the engineering knowledge needed to train and serve MoE models efficiently had become a meaningful part of the gap between frontier labs and second-tier developers.

State-space models were the more ambitious architectural departure. Beginning with Albert Gu and Tri Dao's "Mamba" paper in late 2023, a family of architectures built around structured state-space layers offered an alternative to attention with linear (rather than quadratic) cost in sequence length and a recurrent inference pattern that decoupled compute from context length. Mamba and its successors — Mamba-2, Hyena, RWKV variants —

demonstrated competitive performance on language tasks while offering substantial efficiency advantages for very long contexts and on-device inference.

The state-space line of work did not, in the end, replace the Transformer. By 2025, the dominant pattern at frontier scale was hybrid: a Transformer backbone with occasional state-space layers for certain long-context tasks, or a state-space model with attention layers grafted in. The Jamba model from AI21 was an early and influential example of this hybridisation. The lesson is the same lesson the convolutional-vs-Transformer debate produced in vision: at sufficient scale and engineering investment, the differences between architectural families narrow. The bet on "the next architecture" has, to date, been a worse bet than the bet on "the same architecture, better engineered."

There is one more class of architectural variant worth naming briefly: long-context-specific modifications. Sliding-window attention, ring attention, infinite-context recurrent caches, hierarchical attention — by 2025 the standard context window for frontier production models had grown from 2K tokens (GPT-3) to 200K or 1M tokens, with active research into substantially longer. The techniques are varied and the engineering is intricate, but

the underlying observation is simple: most of the cost of long-context attention is wasteful for most queries, and a great deal of it can be eliminated with the right approximations.

Chapter 9: Multimodality and the Universal Encoder

The Transformer's portability to non-language modalities was one of the field's earlier surprises. The Vision Transformer (ViT) paper from Google in 2020 demonstrated that a near-vanilla Transformer trained on sequences of image patches (a 16×16 patch becoming a token via linear projection) could match or exceed convolutional architectures on ImageNet, provided sufficient training data. Within a year, the ViT recipe had been adopted across video, audio, point clouds, protein structures, and tabular data. The lesson generalised: if you can tokenise the input, you can apply a Transformer to it.

The more consequential development was multimodal alignment — training a single model to process and generate across multiple modalities at once. OpenAI's CLIP (Contrastive Language-Image Pre-training), released in early 2021, trained an image encoder and a text encoder jointly such that paired image-and-caption inputs produced aligned representations. The CLIP embedding space became the substrate for an enormous ecosystem of downstream work: text-to-image generation (DALL-E, Stable Diffusion), zero-shot image classification, image retrieval, and many subsequent multimodal models.

By the end of 2023, most frontier conversational models accepted image inputs natively. GPT-4V, Claude 3, and Gemini 1.5 could all describe images, answer visual questions, and reason about visual content alongside text. The architectural realisation of this varied — some models used separate vision encoders feeding into a shared transformer backbone, others tokenised images directly — but the user-facing capability converged. By 2024, audio input and output were similarly standard. By 2025, video generation models trained as latent-diffusion Transformers (DiT) had reached a quality threshold that allowed minute-long generation at high resolution.

The convergence point of multimodality is what might be called the universal encoder hypothesis: that any modality can be tokenised, that all tokens can be processed by the same Transformer backbone, and that capabilities across modalities will increasingly be jointly trained and jointly improved. Whether this hypothesis turns out to be the right framing or merely a useful interim simplification is one of the open questions of the next several years. It has, however, been a productive working assumption.

A practical note: the engineering challenges of multimodal models are dominated by the asymmetry of data quantity across modalities. Text data is plentiful and cheap; high-quality paired image-and-text data is scarcer and more expensive to produce; high-quality video data is scarcer still. The training recipes that produce strong multimodal models therefore typically involve careful curriculum design, mixing of unimodal and multimodal data, and many small architectural choices about how modality-specific features are fused. The reader interested in this material would do well to start with the Flamingo paper (DeepMind, 2022) and the Florence-2 paper (Microsoft, 2024) as representative landmarks.

Chapter 10: The Inference Frontier

For the first five years of the Transformer's life, ML research was overwhelmingly focused on training. By 2022 it had become apparent that inference — the cost of actually serving these models to users — was the dominant economic problem in deployment. The frontier of inference research over the following years was as productive as any subfield in machine learning, and it deserves a dedicated chapter for the practitioner.

The single most important inference-side contribution of the period was Tri Dao's FlashAttention, published in 2022 and refined through FlashAttention-2 (2023) and FlashAttention-3 (2024). FlashAttention reformulated the attention computation to be IO-aware: rather than materialising the full attention matrix in HBM (high-bandwidth memory), it computed attention in tiles, keeping intermediate values in the much faster on-chip SRAM. The wall-clock speedup, on the same hardware, was between 2× and 4× for typical sequence lengths, and significantly more for very long sequences. FlashAttention is, by 2026, present in essentially every production transformer inference stack.

The second meaningful contribution was the family of techniques that fall under the heading of continuous batching, dominantly associated with Woosuk Kwon and colleagues' vLLM paper from late 2023. The naive way to batch transformer inference is to wait for a full batch of requests to arrive, run them together to completion, and then start the next batch. The throughput of this approach is bad because individual requests in a batch have different output lengths, and the slowest determines the throughput for all. Continuous batching, with PagedAttention to manage the KV-cache memory across variable-length sequences, dramatically improved real-world throughput by allowing requests to enter and leave the batch dynamically.

The third was the long arc of quantisation work. By 2023 it was clear that 8-bit weights produced negligible quality degradation in most language models. By 2024 the frontier was 4-bit weights (with various activation-quantisation schemes), and several techniques — GPTQ, AWQ, GGUF formats — had reached production maturity. By 2025, 2-bit and ternary quantisation schemes were viable for specific deployment scenarios, with the trade-offs starting to bite. Quantisation alone has reduced inference cost by a multiplicative factor across the period.

The combined effect of these three families of work — IO-aware attention kernels, continuous-batching serving stacks, and aggressive quantisation — has been a roughly two-orders-of-magnitude reduction in the cost of serving a token at a given quality level between 2021 and 2026. This number does more than any other to explain why the conversational AI deployment that seemed impossibly expensive in 2022 became cheap enough to bundle into consumer software by 2025.

Worth noting alongside these are the inference-side architectural innovations: speculative decoding (Yaniv Leviathan and colleagues, 2023), which uses a small draft model to generate candidate continuations that a larger verifier model accepts or rejects, often producing 2–3× speedups; multi-query and grouped-query attention, which trade a small quality cost for substantially reduced KV-cache memory; and the various long-context attention approximations mentioned in the previous chapter. The cumulative effect is a inference stack today that is not architecturally what Vaswani described in 2017, even though the underlying model often is.

Chapter 11: Reasoning, Tool Use, and Agents

The capability frontier of language models, by 2024, was no longer about what a model could produce in a single forward pass. It was about what a system built around a model could accomplish over many forward passes, with structured intermediate steps and external tool calls. The shift from single-shot prediction to multi-step orchestration is the dominant capability story of the 2024–2026 period, and it is worth treating it as a distinct chapter.

The first thread was chain-of-thought prompting, introduced by Jason Wei and colleagues at Google in early 2022. The observation was simple and influential: language models prompted to produce a step-by-step reasoning trace before their final answer were more accurate, particularly on multi-step arithmetic, logical inference, and commonsense tasks. By the end of 2022 the technique was standard in evaluation; by 2023 it was standard in production prompting.

The next thread was tool use. The Toolformer paper from Meta (2023) demonstrated that language models could be fine-tuned to know when to invoke external APIs — calculators, search engines, calendar lookups — and how to incorporate the results into their generation. The ReAct paper from Google (2023) generalised this into an explicit Reason-Act-Observe loop. By late 2023, function calling was a first-class feature of every major frontier-model API.

The third thread was the explicit reasoning model. OpenAI's o1, released in late 2024, was the first widely-deployed model trained specifically to allocate inference-time compute to internal reasoning before producing a user-facing answer. DeepSeek's R1, released in early 2026 with full weights and methodology, demonstrated that a comparable reasoning capability could be developed via reinforcement learning from outcome-based rewards, opening up the reasoning paradigm to the open-source community. The class of techniques is collectively referred to as inference-time scaling: trading additional compute at query time for additional quality, in a way analogous to how training-time compute had been the dominant lever in the earlier period.

The fourth and most recent thread is the agentic loop — systems in which a language model takes actions in an environment (browses the web, edits files, runs code, calls APIs) over a long sequence of interactions, with feedback from each step informing the next. The pattern emerged in earnest around 2024 with research demos (AutoGPT, BabyAGI) and matured into production systems in 2025 and 2026: Claude's coding agents, OpenAI's Operator, Google's Mariner and its descendants. The technical challenges of long-horizon agentic systems — credit assignment over multi-step trajectories, robust error recovery, security around tool execution — are at the centre of current research and product development.

Looking back across the four threads, what is striking is that none of them required architectural changes to the underlying Transformer. The model itself, as a sequence-to-sequence function, has remained the same primitive throughout. What has changed is the scaffolding around it: the prompting techniques, the training recipes, the tool integrations, the orchestration loops. The Transformer, in this sense, has become a kind of computational primitive — analogous to the way the database became a primitive in the previous generation of software. The interesting questions are increasingly about what you build on top of it.

Chapter 12: After Attention

It is unusual, perhaps unwise, to write the closing chapter of a book about an active research field. The fair version of the assessment is the one this chapter attempts: a summary of what looks settled, what remains contested, and where the trajectory appears to be pointing.

What is settled, by the time this book goes to press in 2026, is the Transformer's status as the default architecture for general sequence modelling. Ten years after the 2017 paper, no competing architecture has displaced it at the frontier, and the attempts that have come closest — state-space models in particular — have ended up integrated as components within Transformer-based systems rather than replacing them. The probability that the dominant general-purpose architecture in 2030 is a recognisable descendant of the Transformer is, in this author's judgement, very high.

What is contested is whether the dominant approach for capability improvement over the next several years is more pretraining compute, more inference-time compute,

more architectural innovation, or more data-quality engineering. Each of these positions has serious advocates and serious counterarguments. The most defensible position is probably that all four contribute in different ways for different problems; the most useful one is probably to track which one is producing the most surprising results in any given quarter.

What is harder to read is where the field's locus of progress moves once the marginal returns to pretraining scale finally exhaust themselves. The Chinchilla correction bought another four or five years of fruitful scaling. The data-quality work that followed it has bought another two or three. There will, eventually, be a point at which the returns to throwing more compute and more tokens at the standard recipe diminish below the cost of trying. The current frontier labs disagree, mostly privately, about whether that point is two years away or twelve.

The most underrated open problem of the next decade, in this author's view, is the integration of language models with structured memory and persistent state. The current generation of systems is, despite all their capabilities, fundamentally stateless. Every conversation begins from scratch; every long-horizon task carries its history in the context window or in a hand-built retrieval system. The

architectural and methodological work required to give these systems persistent, growing, accessible memory — in a way that does not require recurrence and does not blow up inference cost — is, plausibly, the next paradigm shift.

The most overrated topic of the same period is, in this author's view, model architecture as such. The set of plausible incremental improvements to the Transformer block is broad and the marginal capability gains from any of them is small. The action, at least in the medium term, is in training recipes, data curation, post-training pipelines, multimodal alignment, inference engineering, and the agentic and reasoning scaffolds discussed in Chapter 11. These are less photogenic than a new architecture diagram. They are also, by every measure that matters to deployed systems, where the actual gains have come from for several years now.

Whatever the next decade produces, the decade we have just lived through was an extraordinary one. The Transformer, considered as an artefact, is the most consequential single architecture in the history of machine learning, with the possible exception of the convolutional neural network. The volume of capability it has unlocked, the velocity of improvement it has

supported, and the breadth of problems to which it has been adapted are without precedent in the field. To work on it during this period, even at the edges, has been an unusual professional gift. The people who built the systems described in this book have done remarkable engineering and remarkable science. The people who will build whatever comes next — whether that is more of the same or something genuinely new — have inherited an architectural substrate, and a community of practice, of a kind that has rarely existed in this field before.

The work, as ever, continues.

Acknowledgements

This book benefitted from conversations with too many colleagues to thank individually. Particular gratitude is owed to the engineers and researchers at the major foundation-model laboratories who, over years of conference dinners and offline correspondence, helped me understand what was actually happening behind their public papers. Several of them disagreed strenuously with portions of this account; the residual errors are mine.

The decision to commission this book from Kelford Press was made by a publisher who believed there was room for a technical narrative aimed at working practitioners rather than the more popular accounts that dominate the trade press. Whether the gamble pays off is a question for the reader. I am grateful, regardless, for the latitude.

The decade this book covers is unusual in machine learning for the openness with which most of it was conducted. Almost every architectural decision, training recipe, and empirical result described in these pages was published, in some form, by the laboratory that produced it. That this is now beginning to change is a development the field should mourn.

A Note on Sources

The technical narrative in this book is drawn primarily from the published arXiv literature, with supplementary use of conference proceedings (NeurIPS, ICLR, ICML, ACL), laboratory blog posts, and a small number of off-the-record interviews conducted between 2023 and 2026.

The reader who wishes to follow the original literature should start with the founding papers — Bahdanau, Cho, and Bengio (2014); Vaswani et al. (2017); Devlin et al. (2018); Radford et al. (2018); Brown et al. (2020); Hoffmann et al. (2022); Wei et al. (2022); Dao et al. (2022); Touvron et al. (2023); Kwon et al. (2023); Gu and Dao (2023) — and from each branch out into the citation network. The full bibliography for this volume is hosted at kelfordpress.com/books/the-decade-of-attention/bibliography.

A small number of citations in the text are deliberately informal — for example, the references to "by multiple accounts" in the discussion of ChatGPT's reception inside OpenAI. These reflect interviews conducted under conditions of non-attribution. The author has, in each case, satisfied herself that the underlying claim is well-established among the relevant community of practice, and has tried to flag uncertainty where it remains.

About the Author

Dr. Priya Iyengar is a machine-learning researcher and engineer with two decades in the field. She trained at the University of Cambridge and held postdoctoral positions at the Max Planck Institute for Intelligent Systems and the Mila in Montreal before moving into industrial research. She has worked at three frontier-model laboratories over the period this book covers, contributing to pretraining infrastructure, alignment methodology, and the open-source release of several mid-sized foundation models. She lives in Bengaluru and writes occasionally for *Kelford Press* on questions of ML methodology and research culture.

The Decade of Attention is her first book.