THE AICE LIBRARY

A Short History of Paying Attention

ประวัติศาสตร์ย่อของการใส่ใจ

2014 — 2026

★

For the AICE team — a map of the road already walked, so the road ahead is easier to see.

สำหรับทีม AICE — แผนที่เส้นทางที่เดินผ่านมาแล้ว เพื่อให้มองเห็นเส้นทางข้างหน้าได้ง่ายขึ้น

2026-07-31

A note on sources

Sourced from the AICE Attention Canon (`~/Desktop/AICE/canon/ATTENTION_CANON.md`); a handful of small narrative gaps — 2014, the 2018–2020 scale era, and the 2019 origin of the "efficiency wars" — filled with directly-cited web sources. Technical terms stay in English on purpose; that's house style.

เล่มนี้เรียบเรียงจาก AICE Attention Canon โดยเติมช่องว่างของเรื่องเล่าบางจุด — ปี 2014, ยุคขยายสเกล 2018–2020 และจุดกำเนิดของ “สงครามประสิทธิภาพ” ปี 2019 — ด้วยแหล่งข้อมูลบนเว็บที่อ้างอิงตรง ศัพท์เทคนิคคงไว้เป็นภาษาอังกฤษโดยตั้งใจ ตามธรรมเนียมของบ้านนี้

`Source of truth · ~/Desktop/AICE/canon/ATTENTION_CANON.md`

Contents

สารบัญ

2014

CHAPTER 1

Learning Where to Look

การเรียนรู้ว่าจะมองตรงไหน

EN

Machine translation in 2014 worked by squeezing an entire source sentence into one fixed-size vector, then unpacking it into the target language — and it fell apart on long sentences, because that one vector had to hold everything at once. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio proposed a fix: let the decoder, at each output word, look back and "align" with the specific source words that mattered right then, instead of relying on a single squeezed summary. They called the scoring mechanism that computed this alignment "attention," though the name wouldn't become famous for another three years. The model still used recurrent networks (RNNs) underneath — attention was a patch on top of them, not yet a replacement. But the core idea — let the model choose what to look at, instead of forcing everything through one bottleneck — is the seed every later chapter in this book grows from.

TH

ปีค.ศ. 2014 การแปลภาษาด้วยเครื่องยังทำงานแบบบีบทั้งประโยคต้นทางให้เหลือเป็นเวกเตอร์ก้อนเดียว แล้วค่อยคลี่ออกมาเป็นภาษาปลายทาง — พอประโยคยาวขึ้นมันพังทันที เพราะเวกเตอร์ก้อนเดียวต้องแบกทุกอย่างไว้พร้อมกัน Bahdanau, Cho และ Bengio เสนอทางแก้ว่า ให้ decoder มอง "ย้อนกลับ" ไปหาคำต้นทางที่เกี่ยวข้องตรงจุดนั้นได้เลยในแต่ละคำที่กำลังแปล แทนที่จะพึ่งเวกเตอร์สรุปก้อนเดียว กลไกให้คะแนนว่าคำไหนควรมองนี้แหละที่พวกเขาเรียกว่า "attention" — แม้ชื่อนี้จะยังไม่ดังจนกว่าจะผ่านไปอีก 3 ปี ตอนนั้นโมเดลยังใช้ RNN เป็นแกนหลักอยู่ attention เป็นแค่ของแต่งเสริม ยังไม่ใช่ตัวที่มาแทนที่ทั้งระบบ แต่ไอเดียแกนกลาง

— ให้โมเดลเลือกเองว่าจะมองตรงไหน แทนที่จะยัดทุกอย่างผ่านคอขวดเดียว — คือเมล็ดพันธุ์ที่ทุกบทต่อจากนี้ในเล่มนี้งอกออกมา

EXAMPLE

Translating a 40-word sentence by first memorizing it as one 10-digit number would be absurd — attention lets the translator glance back at the actual words instead, right when needed.

ตัวอย่าง

จะแปลประโยคยาว 40 คำ โดยท่องมันให้เหลือเป็นเลข 10 หลักก่อนคงบ้าไปแล้ว — attention ทำให้ผู้แปลย้อนไปดูคำจริงๆ ได้เลย ตรงจังหวะที่ต้องใช้

2014 · Bahdanau, Cho & Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate" — arXiv:1409.0473

2017

CHAPTER 2

Attention Is All You Need

EN

In 2017, a team at Google (Vaswani et al.) asked a radical question: what if you threw away the recurrent network entirely and built a model out of nothing but attention? Their answer was the Transformer — every token computes a query, a key, and a value, and attends directly to every other token in the sequence, all in parallel, with no step-by-step recurrence at all. That single change unlocked a training speedup that recurrence had quietly been blocking for years, because parallel matrix multiplication is exactly what GPUs are built for. The paper introduced self-attention as the load-bearing mechanism, plus multi-head attention (several attention "views" running side by side) and an encoder-decoder built entirely from these blocks. Almost a decade later, every architecture in this book — no matter how exotic — still gets compared back to this one as the baseline. Nobody has replaced self-attention outright; every later idea wraps it, compresses it, or interleaves it with something cheaper.

TH

ปี 2017 ทีมจาก Google (Vaswani และคณะ) ตั้งคำถามที่ถือว่าสุดโต่งตอนนั้น — ถ้าตัด recurrent network ทิ้งทั้งหมด แล้วสร้างโมเดลขึ้นจาก attention ล้วนๆ ล่ะ จะเป็นยังไง คำตอบคือ Transformer — ทุก token คำนวณ query, key, value ของตัวเอง แล้ว "มอง" ไปยัง token อื่นทุกตัวในลำดับได้โดยตรง พร้อมกันหมดทุกตัว ไม่ต้องไล่ทีละสเต็ปแบบ recurrence อีกต่อไป การเปลี่ยนจุดเดียวนี้ปลดล็อกความเร็วในการเทรนที่ recurrence เคยเป็นคอขวดมานาน เพราะการคูณเมทริกซ์แบบขนานคือสิ่งที่ GPU ถูกออกแบบมาให้ทำอยู่แล้ว เปเปอร์นี้เสนอ self-attention เป็นกลไกหลัก บวกกับ multi-head attention (attention หลาย "มุมมอง" ทำงานคู่ขนาน

กัน) และโครงสร้าง encoder-decoder ที่สร้างจากบล็อกเหล่านี้ล้วนๆ เกือบสิบปีให้หลัง ทุกสถาปัตยกรรมในเล่มนี้ ต่อให้แปลกใหม่แค่ไหน ก็ยังถูกเทียบกลับมาที่ตัวนี้เป็น baseline อยู่ดี ไม่มีใครแทนที่ self-attention ได้เต็มๆ มีแต่ห่อมันไว้ บีบให้เล็กลง หรือสลับใช้ร่วมกับกลไกที่ถูกกว่า

EXAMPLE

Instead of one reader scanning a paragraph strictly in order, imagine 100 readers scanning the whole page at once, each asking a slightly different question — that's roughly multi-head self-attention, one layer at a time.

ตัวอย่าง

แทนที่จะมีคนอ่านคนเดียวไล่อ่านย่อหน้าทีละบรรทัด ลองนึกภาพคน 100 คนกวาดสายตาอ่านทั้งหน้าพร้อมกัน แต่ละคนถามคำถามต่างกันนิดหน่อย — นั่นแหละคือ multi-head self-attention คร่าวๆ ในหนึ่งเลเยอร์

2017 · Vaswani et al., "Attention Is All You Need" — arXiv:1706.03762

2018–2020

CHAPTER 3

Same Recipe, Bigger Kitchen

สูตรเดิม ครัวใหญ่ขึ้น

EN

Once the Transformer existed, the next question wasn't "how does attention work" but "how big can we make this, and what happens." In late 2018, Google's BERT used the Transformer's encoder side, pre-trained on masked-word prediction over huge unlabeled text, and became the new default starting point for almost every language-understanding task. OpenAI pushed the decoder side instead — GPT, then GPT-2 — showing that a Transformer trained simply to predict the next word, at growing scale, started doing things nobody explicitly taught it. By 2020, GPT-3 took that bet to 175 billion parameters and showed a new trick: few-shot learning from nothing but examples in the prompt, no retraining required. Attention itself barely changed in these years — the 2017 architecture mostly held steady. What changed was everything around it: data, parameter count, and compute, scaled by orders of magnitude, revealing that the same mechanism got qualitatively more capable simply by getting bigger.

TH

พอ Transformer เกิดขึ้นแล้ว คำถามต่อไปไม่ใช่ "attention ทำงานยังไง" อีกต่อไป แต่กลายเป็น "จะขยายให้ใหญ่ได้แค่ไหน แล้วจะเกิดอะไรขึ้น" ปลายปี 2018 BERT ของ Google ใช้ฝั่ง encoder ของ Transformer เทรนแบบทายคำที่ถูกปิดไว้จากข้อความมหาศาลที่ไม่มี label แล้วกลายเป็นจุดตั้งต้นมาตรฐานใหม่ของงานด้านภาษาแทบทุกชนิด ฝั่ง OpenAI เดินอีกทาง เลือกฝั่ง decoder แทน — GPT แล้วต่อด้วย GPT-2 — พิสูจน์ว่า Transformer ที่เทรนมาแค่ให้ทายคำถัดไป พอขยายสเกลขึ้นเรื่อยๆ กลับเริ่มทำสิ่งที่ไม่มีใครสอนมันตรงๆ ได้เอง พอถึงปี 2020 GPT-3 เดิมพัน

ไปที่ 175 พันล้านพารามิเตอร์ แล้วโชว์กลเม็ดใหม่คือ few-shot learning เรียนจากแค่ตัวอย่างในพรอมต์ ไม่ต้องเทรนใหม่เลย ช่วงนี้ตัว attention เองแทบไม่เปลี่ยน สถาปัตยกรรมจากปี 2017 ยังนิ่งอยู่ สิ่งที่เปลี่ยนคือทุกอย่างรอบๆ มันต่างหาก — ข้อมูล จำนวนพารามิเตอร์ และ compute ถูกขยายเป็นหลายเท่าตัว จนพิสูจน์ให้เห็นว่ากลไกเดิมเป๊ะๆ เก่งขึ้นแบบก้าวกระโดดได้ แค่เพราะทำให้มันใหญ่ขึ้น

EXAMPLE

Same recipe, bigger kitchen — BERT and GPT-1/2 proved the Transformer recipe worked; GPT-3 proved that scaling the kitchen 100× changed what came out of it, not just how fast.

ตัวอย่าง

สูตรเดิม แค่ครัวใหญ่ขึ้น — BERT กับ GPT-1/2 พิสูจน์ว่าสูตร Transformer ใช้ได้จริง ส่วน GPT-3 พิสูจน์ว่าแค่ขยายครัวให้ใหญ่ขึ้น 100 เท่า ของที่ออกมาจากครัวก็เปลี่ยนไปเลย ไม่ใช่แค่เร็วขึ้น

2018–2020 · Devlin et al., "BERT" — arXiv:1810.04805 · Brown et al., "GPT-3: Language Models are Few-Shot Learners" — arXiv:2005.14165

2019–2021

CHAPTER 4

The Efficiency Wars

สงครามประสิทธิภาพ

EN

Full self-attention costs grow quadratically with sequence length — double the text, quadruple the compute — and by 2019 that ceiling was the obvious next problem to attack. OpenAI's Sparse Transformer opened the wave that April, factorizing the attention matrix so cost grew closer to $n\sqrt{n}$ than n^2. A flood of variants followed in 2020: Reformer approximated attention with hashing, Linformer projected it to low rank, Performer used random-feature kernels, and Longformer/BigBird mixed local windows with a handful of fixed global tokens. Google's own "Long Range Arena" benchmark that same year lined many of these up side by side on long-sequence tasks, and the results were sobering — most saved real memory and speed, but at a meaningful quality cost, and none became the new default. The industry's real bottleneck, it would later turn out, wasn't FLOPs at all — it was memory movement, a problem this entire generation of papers wasn't built to solve. So most of this generation quietly stopped being used at frontier scale, even though the papers are still cited constantly as the historical record of what got tried. The lesson that survived: approximating the attention matrix is one lever, but it's not the only — or even the best — one.

> TH
>
> ต้นทุนของ self-attention เต็มรูปแบบโตแบบยกกำลังสองตามความยาวของ sequence — ข้อความยาวขึ้น 2 เท่า compute ต้องเพิ่ม 4 เท่า พอถึงปี 2019 เพดานนี้ก็ชัดเจนแล้วว่าเป็นปัญหาถัดไปที่ต้องแก้ Sparse Transformer ของ OpenAI เปิดคลื่นลูกนี้ตั้งแต่เดือนเมษายนปีนั้น ด้วยการแยกตัว attention matrix

ให้ต้นทุนใกล้เคียง n√n มากกว่า n² ปี 2020 มีตัวแปรพรั่งพรูตามมาเพียบ — Reformer ใช้ hashing มาประมาณค่า attention, Linformer โปรเจกต์ลง low rank, Performer ใช้ random-feature kernel ส่วน Longformer กับ BigBird ผสม local window เข้ากับ global token จำนวนหนึ่ง เบนช์มาร์ก "Long Range Arena" ของ Google ปีเดียวกัน เอาโมเดลพวกนี้มาเทียบกันแบบเห็นภาพชัดบนงาน sequence ยาวๆ ผลออกมาค่อนข้างจริงจัง — ส่วนใหญ่ประหยัดความจำและเร็วขึ้นจริง แต่แลกกับคุณภาพที่หายไปพอสมควร ไม่มีตัวไหนกลายเป็นค่าเริ่มต้นใหม่ ปัญหาคอขวดจริงของอุตสาหกรรม ซึ่งจะรู้กันชัดในอีกไม่กี่ปีถัดมา กลับไม่ใช่ FLOPs เลย แต่เป็นการขนย้ายข้อมูลในหน่วยความจำ ซึ่งเป็นโจทย์ที่เปเปอร์ยุคนี้ทั้งรุ่นไม่ได้ถูกออกแบบมาแก้ โมเดลกลุ่มนี้ส่วนใหญ่จึงค่อยๆ เลิกถูกใช้ในระดับ frontier แม้เปเปอร์จะยังถูกอ้างอิงตลอดในฐานะบันทึกว่าไอเดียไหนเคยถูกลองมาแล้วบ้าง บทเรียนที่เหลือรอดมาคือ การประมาณค่า attention matrix เป็นแค่คันโยกหนึ่ง ไม่ใช่คันโยกเดียว หรือด้วยซ้ำคันโยกที่ดีที่สุด

EXAMPLE

A dozen inventors race to build a cheaper engine by removing parts — several succeed at "cheaper," almost none keep "drives just as well," and the real bottleneck turns out to be traffic (memory bandwidth), not the engine (FLOPs).

ตัวอย่าง

นักประดิษฐ์สิบกว่าคนแข่งกันสร้างเครื่องยนต์ให้ถูกลงด้วยการถอดชิ้นส่วนออก หลายคนทำได้ "ถูกลง" จริง แต่แทบไม่มีใครรักษา "วิ่งดีเหมือนเดิม" ไว้ได้ แล้วคอขวดจริงกลับกลายเป็นรถติด (แบนด์วิดท์หน่วยความจำ) ไม่ใช่ตัวเครื่องยนต์ (FLOPs)

2019–2021 · Child, Gray, Radford & Sutskever, "Sparse Transformer" — arXiv:1904.10509 · Tay et al., "Long Range Arena" — arXiv:2011.04006

2022

CHAPTER 5

FlashAttention: The Hardware Turn

จุดเปลี่ยนที่ฮาร์ดแวร์

EN

By 2022, the field's efficiency instinct changed direction entirely. Instead of approximating attention to make it cheaper, Tri Dao and colleagues asked a different question: what if we compute the exact same attention, with zero quality loss, just smarter about where the numbers live during the computation? Their answer, FlashAttention, tiles the computation so it stays inside a GPU's small, fast on-chip memory (SRAM) instead of constantly writing and reading the full attention matrix from slower high-bandwidth memory. The result was a large speedup with no approximation at all — free performance, which is rare in this field. Within a couple of years it became the single most-deployed efficiency trick in modern AI infrastructure, quietly running underneath almost every serious training or inference job on NVIDIA hardware. FlashAttention has since gone through three more generations (FA2, FA3, FA4), each tuned to newer GPU hardware — but it has never run natively on Apple Silicon, a gap that matters later in this book.

TH

พอถึงปี 2022 สัญชาตญาณเรื่องประสิทธิภาพของวงการก็เปลี่ยนทิศไปเลย แทนที่จะประมาณค่า attention ให้ถูกลง Tri Dao กับทีมตั้งคำถามใหม่ — ถ้าเราคำนวณ attention แบบเป๊ะๆ เหมือนเดิมทุกอย่าง ไม่เสียคุณภาพเลยสักนิด แค่ฉลาดขึ้นเรื่องว่าตัวเลขควรอยู่ตรงไหนระหว่างคำนวณล่ะ คำตอบของพวกเขาคือ FlashAttention ซึ่ง tile การคำนวณให้อยู่ในหน่วยความจำเร็วบนชิป GPU (SRAM) แทนที่จะเขียนอ่าน attention matrix เต็มตัวจากหน่วยความจำแบนด์วิดท์สูงที่ช้ากว่าตลอดเวลา ผลลัพธ์คือเร็วขึ้นมากแบบไม่ต้องประมาณค่าอะไรเลย เป็น

> performance ที่ได้มาฟรีๆ ซึ่งหาได้ยากมากในวงการนี้ ภายในไม่กี่ปี มันกลายเป็นเทคนิคด้านประสิทธิภาพที่ถูกใช้งานกว้างขวางที่สุดในโครงสร้างพื้นฐาน AI สมัยใหม่ วิ่งอยู่เงียบๆ เบื้องหลังงานเทรนหรือ inference จริงจังแทบทุกงานบนฮาร์ดแวร์ NVIDIA จากนั้น FlashAttention พัฒนาต่ออีก 3 รุ่น (FA2, FA3, FA4) แต่ละรุ่นปรับจูนให้เข้ากับฮาร์ดแวร์ GPU รุ่นใหม่ๆ แต่ไม่เคยรันแบบเนทีฟบน Apple Silicon เลยสักรุ่น — ช่องว่างนี้สำคัญมากในบทหลังๆ ของเล่มนี้

EXAMPLE

It's the difference between re-fetching one ingredient from a far pantry for every single step of a recipe, versus laying out everything needed on the counter first — same dish, same recipe, far less walking.

ตัวอย่าง

เหมือนความต่างระหว่างการวิ่งไปหยิบวัตถุดิบจากตู้ไกลๆ ทีละขั้นตอนของสูตรอาหาร กับการจัดทุกอย่างที่ต้องใช้มาวางบนเคาน์เตอร์ไว้ก่อนเลย — จานเดิม สูตรเดิม แต่เดินน้อยลงเยอะ

2022 · Dao et al., "FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness," NeurIPS — arXiv:2205.14135

2023

CHAPTER 6

Serving It for Real

เสิร์ฟของจริงให้ได้

EN

By 2023, the field had a new problem: attention worked, and scaling worked, but actually serving these models to real users at long context was its own unsolved engineering challenge. Four separate fixes landed the same year, each attacking a different part of the problem. Grouped-Query Attention (GQA) let many query heads share just a couple of key/value heads, shrinking the KV cache that grows with every generated token. PagedAttention (used in vLLM) stopped treating that KV cache as one giant reserved memory block and instead paged it the way an operating system manages RAM, nearly eliminating waste. StreamingLLM discovered a model could generate forever over a live stream if it kept a small handful of "sink" tokens from the very start, alongside a sliding window of recent ones. Ring Attention showed how to split a sequence across many machines for training on contexts too long for any single accelerator's memory. None of these changed what attention computes — they changed what it costs to actually run, which turned out to be the more urgent problem.

TH

พอถึงปี 2023 วงการเจอปัญหาใหม่ — attention ใช้งานได้ดี การขยายสเกลก็ได้ผล แต่การเสิร์ฟโมเดลพวกนี้ให้ผู้ใช้จริงที่ context ยาวๆ กลับเป็นโจทย์วิศวกรรมอีก ก้อนที่ยังไม่มีใครแก้ได้ ปีเดียวกันนั้นมีทางแก้ 4 อย่างลงพร้อมกัน แต่ละอย่างจัดการคนละส่วนของปัญหา Grouped-Query Attention (GQA) ให้ query head หลายตัวใช้ key/value head ร่วมกันแค่ไม่กี่ตัว ทำให้ KV cache ที่โตขึ้นทุก token ที่สร้างมีขนาดเล็กลง PagedAttention (ที่ใช้ใน vLLM) เลิกมองว่า KV cache เป็นบล็อก

ความจำก้อนใหญ่ที่ต้องจองไว้ล่วงหน้า แล้วเปลี่ยนมาแบ่งเป็นหน้าๆ แบบที่ OS จัดการ RAM แทน ลดความสูญเปล่าไปเกือบหมด StreamingLLM ค้นพบว่าโมเดลสร้างข้อความต่อไปเรื่อยๆ ไม่มีที่สิ้นสุดได้ ถ้าเก็บ token "sink" หยิบมือหนึ่งจากจุดเริ่มต้นไว้เสมอ คู่กับ sliding window ของ token ล่าสุด ส่วน Ring Attention โชว์วิธีตัดแบ่ง sequence ข้ามหลายเครื่อง สำหรับเทรนบน context ที่ยาวเกินกว่าหน่วยความจำของการ์ดใบเดียวจะรับไหว ไม่มีอันไหนเปลี่ยนสิ่งที่ attention คำนวณเลย มีแต่เปลี่ยนต้นทุนของการรันจริง ซึ่งกลายเป็นปัญหาที่เร่งด่วนกว่าด้วยซ้ำ

EXAMPLE

Four plumbers fix the same house's water bill the same year — one shares pipes between rooms (GQA), one stops reserving a full tank per room (PagedAttention), one keeps a tap open so pressure never collapses (attention sinks), one splits the house across a street of houses (Ring Attention).

ตัวอย่าง

ช่างประปา 4 คน แก้บิลค่าน้ำบ้านเดียวกันในปีเดียวกัน — คนหนึ่งแชร์ท่อระหว่างห้อง (GQA) คนหนึ่งเลิกจองถังน้ำเต็มใบต่อห้อง (PagedAttention) คนหนึ่งเปิดก๊อกทิ้งไว้นิดหนึ่งไม่ให้แรงดันตก (attention sinks) อีกคนตัดแบ่งบ้านกระจายไปทั้งซอย (Ring Attention)

2023 · Ainslie et al., "GQA" — arXiv:2305.13245 · Kwon et al., "PagedAttention / vLLM" — vllm.ai blog · Xiao et al., "StreamingLLM" — arXiv:2309.17453 · Liu, Zaharia & Abbeel, "Ring Attention" — arXiv:2310.01889

2023–2024

CHAPTER 7

Mixture-of-Experts Goes Mainstream

MoE กลายเป็นกระแสหลัก

EN

Mixture-of-Experts wasn't new in 2023 — Noam Shazeer had proposed sparsely-gated MoE back in 2017, and Google's Switch Transformer simplified it to routing each token to just one expert in 2021. What changed in 2023–2024 was adoption at the frontier: instead of one dense feed-forward block per layer doing all the work, labs began shipping models with many specialized expert blocks and a small router sending each token to only a handful of them. The appeal is a kind of economic trick — total parameter count (and therefore capability) can be enormous, while the compute actually spent per token stays small, because most experts sit idle for any given token. The remaining hard problem was routing: how do you keep the router from collapsing onto a few favorite experts, without distorting the model's actual training goal to force balance? DeepSeek's 2024 answer — an auxiliary-loss-free approach, where a learnable per-expert bias nudges routing based on observed load but stays outside the actual gating-weight math — became the practice other labs converged on. By 2026, the pattern is universal: every major frontier release is MoE; the question moved from "MoE or not" to "how do you route well."

> TH
>
> Mixture-of-Experts ไม่ใช่ของใหม่ในปี 2023 — Noam Shazeer เสนอ sparsely-gated MoE ไว้ตั้งแต่ปี 2017 แล้ว และ Switch Transformer ของ Google ก็ทำให้มันเรียบง่ายขึ้นด้วยการส่ง token แต่ละตัวไปหา expert แค่ตัวเดียวในปี 2021 สิ่งที่

เปลี่ยนในปี 2023–2024 คือการถูกนำไปใช้จริงในระดับ frontier แทนที่จะมี feed-forward block หนาแน่นตัวเดียวต่อเลเยอร์ทำงานทั้งหมด แล็บต่างๆ เริ่มปล่อยโมเดลที่มี expert block เฉพาะทางจำนวนมาก พร้อม router ตัวเล็กๆ ที่ส่ง token แต่ละตัวไปหาแค่หยิบมือเดียวจากทั้งหมด เสน่ห์ของมันเหมือนกลเม็ดทางเศรษฐศาสตร์ — จำนวนพารามิเตอร์รวม (และความสามารถ) ใหญ่มหาศาลได้ ในขณะที่ compute ที่ใช้จริงต่อ token ยังเล็กอยู่ เพราะ expert ส่วนใหญ่ "ว่างงาน" สำหรับ token ใดๆ ก็ตาม ปัญหาที่ยังเหลือคือเรื่อง routing — จะป้องกันไม่ให้ router ไปกองอยู่ที่ expert คนโปรดไม่กี่ตัวได้ยังไง โดยไม่บิดเบือนเป้าหมายการเทรนจริงของโมเดลเพื่อบังคับความสมดุล คำตอบของ DeepSeek ในปี 2024 คือแนวทาง auxiliary-loss-free ที่ให้ bias ต่อ expert แต่ละตัวที่เรียนรู้ได้ คอยปรับ routing ตามภาระงานที่สังเกตเห็นจริง แต่อยู่นอกสมการคำนวณน้ำหนัก gating จริงๆ กลายเป็นแนวทางที่แล็บอื่นๆ ตามมาบรรจบกันหมด พอถึงปี 2026 แพทเทิร์นนี้กลายเป็นสากล — ทุกโมเดล frontier รุ่นใหญ่เป็น MoE หมด คำถามเปลี่ยนจาก "จะใช้ MoE ไหม" ไปเป็น "จะ route ยังไงให้ดี" แทน

EXAMPLE

A hospital with hundreds of specialists on staff, but any single patient only ever sees a small handful of them, plus one generalist always on duty — huge total expertise, small cost per visit.

ตัวอย่าง

โรงพยาบาลที่มีผู้เชี่ยวชาญหลายร้อยคนในทีม แต่คนไข้แต่ละคนได้เจอแค่หยิบมือเดียวจากทั้งหมด บวกกับหมอทั่วไปประจำอีกหนึ่งคน ความเชี่ยวชาญรวมมหาศาล แต่ต้นทุนต่อการมาหาหมอหนึ่งครั้งกลับเล็กนิดเดียว

2023–2024 · Shazeer et al. — arXiv:1701.06538 · Fedus et al., "Switch Transformer" — arXiv:2101.03961 · DeepSeek-V3 auxiliary-loss-free balancing — arXiv:2408.15664

2024

CHAPTER 8

DeepSeek's Compression Trick (MLA)

กลเม็ดบีบอัดของ DeepSeek

EN

In May 2024, DeepSeek shipped an idea aimed squarely at the KV-cache problem GQA had only partly solved. Instead of sharing key/value heads across query heads, Multi-Head Latent Attention (MLA) compresses the keys and values for each token jointly into one small low-rank latent vector, caches only that tiny vector, and reconstructs the full keys/values back out at attention time. A companion trick — a decoupled rotary position scheme — makes sure positional information survives being compressed this way. The reported result was startling for a compression technique: MLA didn't just save memory, it matched or beat plain multi-head attention on quality, while cutting KV-cache size by roughly 7 to 14 times. That combination — smaller and better, not smaller-but-worse — is unusual enough that MLA quickly spread beyond DeepSeek, into Kimi and GLM's later model lines. It's a different lever from GQA (fewer heads) or hybrid linear attention (Logy's own approach) — proof there was more than one valid way to attack the same cost problem.

TH

เดือนพฤษภาคม 2024 DeepSeek ปล่อยไอเดียที่เล็งไปตรงปัญหา KV-cache ที่ GQA แก้ได้แค่ครึ่งเดียว แทนที่จะแชร์ key/value head ข้าม query head แบบ GQA, Multi-Head Latent Attention (MLA) บีบอัด key/value ของแต่ละ token รวมกันให้เหลือเป็น latent vector เล็กๆ ก้อนเดียวแบบ low-rank เก็บแค่เวกเตอร์จิ๋วนั้นไว้ใน cache แล้วค่อยขยายกลับเป็น key/value เต็มตอนคำนวณ attention

จริง มีเทคนิคคู่กันคือ decoupled rotary position scheme ที่ทำให้ข้อมูลตำแหน่งยังรอดอยู่แม้จะถูกบีบแบบนี้ ผลลัพธ์ที่รายงานออกมาน่าประหลาดใจสำหรับเทคนิคบีบอัด — MLA ไม่ได้แค่ประหยัดความจำ แต่คุณภาพยังเท่าหรือดีกว่า plain multi-head attention ด้วย ในขณะที่ลดขนาด KV-cache ลงราว 7–14 เท่า การได้ทั้งเล็กลงและดีขึ้น ไม่ใช่เล็กลงแต่แย่ลง เป็นอะไรที่แปลกพอจะทำให้ MLA แพร่ไปเร็วเกินขอบเขต DeepSeek เข้าไปอยู่ในโมเดลรุ่นหลังของ Kimi และ GLM ด้วย มันเป็นคันโยกคนละแบบกับ GQA (ลดจำนวน head) หรือ hybrid linear attention (แนวทางของ Logy เอง) — เป็นหลักฐานว่ามีมากกว่าหนึ่งวิธีที่ใช้ได้จริงในการโจมตีปัญหาต้นทุนเดียวกันนี้

EXAMPLE

Instead of two people sharing one notebook (GQA), everyone writes in shorthand, and a translator expands it back to full sentences only when someone actually needs to read it (MLA).

ตัวอย่าง

แทนที่จะมีสองคนแชร์สมุดจดเล่มเดียวกัน (GQA) ทุกคนเขียนแบบชวเลข แล้วมีนักแปลคอยขยายกลับเป็นประโยคเต็มเฉพาะตอนที่มีคนต้องอ่านจริงๆ (MLA)

2024 · DeepSeek-V2, "MLA" — arXiv:2405.04434 · note: Logy does not use MLA (canon §4) — its savings come from GQA + hybrid linear attention instead (see Ch.12)

2025

CHAPTER 9

Trainable Sparsity & Linear Attention's Comeback

Sparsity ที่เทรนมาแต่ต้น กับการคัมแบ็กของ Linear Attention

EN

2025 made sparsity and linear attention production-ready instead of merely clever. Native Sparse Attention (NSA), from DeepSeek, trains a model from scratch with three parallel attention branches — coarse compressed blocks, learned fine-grained selected blocks, and a local sliding window — instead of bolting sparsity onto an already-trained dense model afterward. The same month, Moonshot's MoBA applied the Mixture-of-Experts principle directly to attention: a learned router picks which blocks of the sequence each query actually needs to see. Both won major recognition that year — NSA an ACL Best Paper, MoBA a NeurIPS spotlight — and both share one lesson: efficiency trained in from scratch beats efficiency bolted on after. In parallel, a second lineage matured: Gated DeltaNet, a hybrid linear-attention layer combining gated decay with an error-correcting "delta rule," paired at a 3-to-1 ratio with ordinary full attention wrapped in a new sigmoid gate — the same gate that, not coincidentally, directly answers the attention-sink problem StreamingLLM first flagged back in 2023. This exact pairing — three parts Gated DeltaNet to one part gated full attention — is the lineage Logy is built on.

> TH
>
> ปี 2025 ทำให้ sparsity กับ linear attention "ใช้งานจริงได้" ไม่ใช่แค่ฉลาดในเปเปอร์อีกต่อไป Native Sparse Attention (NSA) จาก DeepSeek เทรนโมเดลตั้งแต่ต้นด้วย attention 3 สาขาคู่ขนาน — บล็อกหยาบที่บีบอัดไว้ บล็อกละเอียดที่

เลือกแบบเรียนรู้ได้ และ local sliding window — แทนที่จะเอา sparsity ไปแปะทับโมเดล dense ที่เทรนเสร็จแล้วทีหลัง เดือนเดียวกันนั้น MoBA ของ Moonshot เอาหลักการ Mixture-of-Experts มาใช้กับตัว attention ตรงๆ เลย — router ที่เรียนรู้ได้เลือกว่า query แต่ละตัวควรเห็นบล็อกไหนของ sequence บ้าง ทั้งคู่ได้รางวัลใหญ่ในปีนั้น — NSA ได้ ACL Best Paper ส่วน MoBA ได้ NeurIPS spotlight — และให้บทเรียนเดียวกันคือ ประสิทธิภาพที่ฝังมาตั้งแต่เทรน ดีกว่าประสิทธิภาพที่แปะเพิ่มทีหลัง ในเวลาเดียวกัน อีกสายพันธุ์หนึ่งก็เติบโตเต็มที่ คือ Gated DeltaNet เลเยอร์ linear-attention แบบ hybrid ที่รวม gated decay กับ "delta rule" คอยแก้ error จับคู่ในอัตราส่วน 3 ต่อ 1 กับ full attention ธรรมดาที่ห่อด้วย sigmoid gate ตัวใหม่ — ซึ่งไม่ใช่เรื่องบังเอิญที่ gate ตัวนี้แหละที่ตอบโจทย์ปัญหา attention-sink ที่ StreamingLLM เคยชี้ไว้ตั้งแต่ปี 2023 การจับคู่แบบนี้เป๊ะๆ — Gated DeltaNet 3 ส่วน ต่อ gated full attention 1 ส่วน — คือสายพันธุ์ที่ Logy ถูกสร้างขึ้นมาจากมัน

EXAMPLE

NSA/MoBA are like teaching an intern the shortcut on day one of training, instead of hiring a fully-trained expert and asking them to unlearn old habits and take shortcuts later.

ตัวอย่าง

NSA/MoBA เหมือนสอนทางลัดให้เด็กฝึกงานตั้งแต่วันแรกที่เข้ามาเทรน แทนที่จะจ้างผู้เชี่ยวชาญที่เทรนมาเต็มรูปแบบแล้ว มาขอให้เขาลืมนิสัยเดิมแล้วหัดลัดทีหลัง

2025 · DeepSeek/Peking University/U. Washington, "NSA" — arXiv:2502.11089 · Moonshot AI/Tsinghua/Zhejiang, "MoBA" — arXiv:2502.13189 · Gated DeltaNet origin — arXiv:2412.06464 · gating fix, NeurIPS 2025 Best Paper — arXiv:2505.06708

2026

CHAPTER 10

Four Labs, Four Bets

สี่แล็บ สี่เดิมพัน

EN

By early 2026, attention architecture stopped converging and started diverging again. Within roughly a two-week window in February, four frontier labs shipped four different, incompatible bets on the same underlying problem. GLM-5 combined MLA with DeepSeek Sparse Attention; Kimi K2.5 used MLA alone, without a sparse-attention layer; Qwen3.5 (and its April successor Qwen3.6 — Logy's own lineage) bet on the hybrid Gated-DeltaNet-plus-Gated-Attention shape. And MiniMax's M2.5 did something nobody expected from a lab that had championed linear attention since 2025: it reverted to plain, full multi-head attention, reportedly "for reliability." No single scheme won outright — each is a different trade-off between memory, quality, training cost, and serving simplicity. It's a useful reminder for anyone reading a "settled best practice" claim about attention: in this field, settled claims have a short half-life, and simplicity is still, sometimes, the winning bet.

TH

พอถึงต้นปี 2026 สถาปัตยกรรม attention ก็เลิกลู่เข้าหากัน แล้วเริ่มแตกทางอีกครั้ง ในช่วงเวลาสั้นๆ ราวสองสัปดาห์ของเดือนกุมภาพันธ์ แล็บ frontier 4 เจ้าปล่อยของพร้อมกัน 4 ทางเลือกที่ต่างกันและเข้ากันไม่ได้ สำหรับปัญหาเดียวกัน เป๊ะๆ GLM-5 รวม MLA เข้ากับ DeepSeek Sparse Attention, Kimi K2.5 ใช้ MLA ล้วนๆ ไม่มีเลเยอร์ sparse attention เลย, ส่วน Qwen3.5 (และรุ่นต่อยอดเดือนเมษายน Qwen3.6 ที่เป็นสายพันธุ์เดียวกับ Logy) เดิมพันกับรูปแบบ hybrid Gated-DeltaNet บวก Gated-Attention และ M2.5 ของ MiniMax ทำสิ่งที่ไม่มีใครคาดคิด

จากแล็บที่เชียร์ linear attention มาตั้งแต่ปี 2025 — มันถอยกลับไปใช้ plain full multi-head attention ธรรมดา โดยให้เหตุผลว่า "เพื่อความน่าเชื่อถือ" ไม่มีสูตรไหนชนะขาดลอย แต่ละตัวคือการแลกเปลี่ยนคนละแบบระหว่างความจำ คุณภาพ ต้นทุนการเทรน และความง่ายในการเสิร์ฟ เป็นเครื่องเตือนใจที่ดีสำหรับใครก็ตามที่เจอคำกล่าวอ้างว่า "นี่คือ best practice ที่ลงตัวแล้ว" เกี่ยวกับ attention — ในวงการนี้ คำกล่าวอ้างแบบ "ลงตัวแล้ว" มีอายุสั้นมาก และความเรียบง่ายบางที ก็ยังเป็นเดิมพันที่ชนะได้อยู่ดี

EXAMPLE

Four chefs get the same ingredients and two weeks, and come back with four completely different dishes — one of them, famous for fusion cooking, serves plain rice and calls it the safer bet.

ตัวอย่าง

เชฟ 4 คนได้วัตถุดิบชุดเดียวกันกับเวลาสองสัปดาห์เท่ากัน กลับมาพร้อมอาหาร 4 จานที่ต่างกันโดยสิ้นเชิง คนหนึ่งซึ่งดังเรื่องอาหารฟิวชั่น กลับเสิร์ฟข้าวสวยเปล่าๆ แล้วบอกว่านี่แหละคือทางเลือกที่ปลอดภัยกว่า

2026 · Labonne, "Qwen3.5" write-up — huggingface.co/blog/mlabonne/qwen35

CHAPTER 11

The Future Five

ห้าอนาคตที่กำลังจะมาถึง

EN

Five more ideas are alive in research right now, none of them yet the default in a shipped frontier chat model — worth knowing not because they're mainstream, but because one of them plausibly is Logy's next chapter.

Adaptive Attention — let a model spend more compute on hard tokens and less on easy ones, instead of a fixed budget for every token; still research-stage, no shipped flagship uses it yet.

Structured Attention — impose a designed sparsity pattern (block, graph, dilated) instead of letting the model learn where to look, trading flexibility for a guaranteed worst-case cost.

Memory-Augmented Attention — give a model a separate, persistent memory beyond its attention window; Titans goes furthest, updating its own weights while reading, scaling past 2 million tokens.

Multi-Modal Attention — already not the future but the present: Logy itself ships a vision encoder trained in from scratch, not bolted on afterward.

Attention Compression — the strangest one: DeepSeek-OCR renders old text context as an image and reads it back with a cheap vision encoder, for roughly 10–20× fewer tokens carrying the same information.

> TH
>
> อีกห้าไอเดียที่ยังมีชีวิตอยู่ในงานวิจัยตอนนี้ ยังไม่มีอันไหนกลายเป็นค่าเริ่มต้นของแชทโมเดล frontier ตัวไหนที่ปล่อยจริงเลย — ควรรู้ไว้ไม่ใช่เพราะมันเมนสตรีมแล้ว แต่เพราะมีอันหนึ่งที่มีโอกาสสูงว่าจะเป็นบทต่อไปของ Logy เอง

Adaptive Attention — ให้โมเดลทุ่ม compute กับ token ที่ยากมากขึ้น และใช้น้อยลงกับ token ที่ง่าย แทนที่จะให้งบเท่ากันหมดทุก token ตอนนี้ยังอยู่ในขั้นวิจัย ยังไม่มีโมเดลเรือธงตัวไหนใช้จริง

Structured Attention — กำหนด sparsity pattern แบบตายตัวไว้ก่อน (block, graph, dilated) แทนที่จะให้โมเดลเรียนรู้เองว่าควรมองตรงไหน แลกความยืดหยุ่นกับต้นทุนขั้นแย่สุดที่รับประกันได้ล่วงหน้า

Memory-Augmented Attention — ให้โมเดลมีหน่วยความจำแยกต่างหากที่อยู่ได้นานกว่าหน้าต่าง attention ปกติ Titans ไปไกลสุด ปรับน้ำหนักตัวเองระหว่างอ่านเลย ไปได้ไกลกว่า 2 ล้าน token

Multi-Modal Attention — จริงๆ ไม่ใช่ "อนาคต" แล้ว แต่เป็น "ปัจจุบัน" — Logy เองมี vision encoder ที่เทรนมาตั้งแต่ต้นด้วยกัน ไม่ใช่ต่อเติมทีหลัง

Attention Compression — ตัวที่แปลกที่สุด DeepSeek-OCR เรนเดอร์ context ข้อความเก่าให้เป็นภาพ แล้วอ่านกลับด้วย vision encoder ราคาถูก ได้ token น้อยลงราว 10–20 เท่า สำหรับข้อมูลชุดเดียวกัน

EXAMPLE

DeepSeek-OCR is like photographing a page of notes instead of retyping it word-for-word — the photo (image tokens) holds almost all the same information in a fraction of the storage.

ตัวอย่าง

DeepSeek-OCR เหมือนถ่ายรูปหน้าโน้ตแทนที่จะพิมพ์มันใหม่ทีละคำ — รูปถ่าย (image tokens) เก็บข้อมูลได้เกือบเท่าเดิม แต่ใช้พื้นที่แค่เศษเสี้ยวเดียว

2019–2025 · Sukhbaatar et al. (Adaptive Attention precursor) — arXiv:1905.07799 · Behrouz et al., "Titans" — arXiv:2501.00663 · Qwen3.5 multimodal — HF blog, Feb 2026 · DeepSeek-OCR — arXiv:2510.18234

CHAPTER 12

Inside Logy

ข้างในตัว Logy

EN

Every idea in this book converges, right now, inside the model running on ICE's own Mac. Logy — Qwen3.6-35B-A3B — has 40 transformer layers arranged as ten repeating blocks; in each block, three layers run Gated DeltaNet (2024's hybrid linear attention, the direct descendant of the RNN this whole story started by trying to replace) and one layer runs "Gated Attention" — ordinary GQA-style full attention (16 query heads sharing just 2 key/value heads) wrapped in a 2025 sigmoid gate that fights the exact attention-sink problem StreamingLLM first flagged back in 2023. Wrapped around every one of those 40 layers sits a Mixture-of-Experts feed-forward block: 256 experts total, but only 8 routed plus 1 shared expert actually switched on per token — the same idea Shazeer proposed in 2017, now load-bearing. That combination of mostly-constant-cost recurrent layers and a small handful of active experts is very likely why a native 262,144-token context fits in one Mac's unified memory at all, extensible past a million tokens via RoPE scaling. Some finer details aren't publicly confirmed for this exact checkpoint: most likely the DeltaNet gate still works per-head rather than per-channel, consistent with its own lineage, and it's unclear whether QK-Norm survives into these layers — both marked **[inferred]**, not stated as settled fact. Logy does not use MLA, NSA, MoBA, or Ring Attention — its whole efficiency story is "hybrid linear attention + GQA + MoE," nothing more exotic than that. Logy doesn't just use attention — nine years of the field's argument about how to make attention affordable is, quite literally, sitting in its weights.

TH

ทุกไอเดียในเล่มนี้มาบรรจบกัน ตอนนี้ ในโมเดลที่รันอยู่บน Mac ของ ICE เอง Logy หรือ Qwen3.6-35B-A3B มี 40 เลเยอร์ Transformer จัดเป็น 10 บล็อกที่ซ้ำกัน ในแต่ละบล็อก 3 เลเยอร์รัน Gated DeltaNet (hybrid linear attention ของปี 2024 ซึ่งเป็นทายาทตรงของ RNN ที่เรื่องทั้งหมดนี้เริ่มต้นจากการพยายามแทนที่มัน) และอีก 1 เลเยอร์รัน "Gated Attention" — GQA แบบปกติ (query head 16 ตัวใช้ key/value head ร่วมกันแค่ 2 ตัว) ห่อด้วย sigmoid gate จากปี 2025 ที่สู้กับปัญหา attention-sink ตัวเดียวกับที่ StreamingLLM เคยชี้ไว้ตั้งแต่ปี 2023 ห่อรอบทั้ง 40 เลเยอร์นี้คือ Mixture-of-Experts feed-forward block — expert รวม 256 ตัว แต่เปิดใช้งานจริงแค่ 8 ตัวที่ routed บวก 1 ตัวที่ shared ต่อ token เดียว ไอเดียเดียวกับที่ Shazeer เสนอไว้ตั้งแต่ปี 2017 ตอนนี้กลายเป็นแกนสำคัญไปแล้ว การผสมกันของเลเยอร์ recurrent ที่ต้นทุนเกือบคงที่ กับ expert หยิบมือเดียวที่ทำงานจริง น่าจะเป็นเหตุผลหลักที่ context ยาวถึง 262,144 token แบบเนทีฟ ยัดลงในหน่วยความจำ unified ของ Mac เครื่องเดียวได้ ขยายไปได้ไกลกว่าล้าน token ด้วย RoPE scaling รายละเอียดปลีกย่อยบางจุดยังไม่มีแหล่งข้อมูลสาธารณะยืนยันสำหรับ checkpoint ตัวนี้เป๊ะๆ — น่าจะเป็นไปได้มากว่า gate ของ DeltaNet ยังทำงานแบบต่อ-head ไม่ใช่ต่อ-channel ตามสายพันธุ์เดิมของมัน และก็ยังไม่ชัดว่า QK-Norm รอดมาถึงเลเยอร์พวกนี้ไหม — ทั้งคู่ขอทำเครื่องหมาย **[inferred]** ไว้ ไม่ใช่ข้อเท็จจริงที่ฟันธงได้ Logy ไม่ใช้ MLA, NSA, MoBA หรือ Ring Attention — เรื่องประสิทธิภาพทั้งหมดของมันคือ "hybrid linear attention + GQA + MoE" ไม่มีอะไรซับซ้อนไปกว่านั้น Logy ไม่ได้แค่ "ใช้" attention เท่านั้น — เก้าปีของข้อถกเถียงทั้งวงการว่าจะทำให้ attention ใช้งานได้จริงในราคาที่จ่ายไหว ตอนนี้นั่งอยู่ในน้ำหนักของมันเอง ตามตัวอักษรเลย

EXAMPLE

Open Activity Monitor while asking Logy a long question — the near-flat memory curve you'd see is 30 of its 40 layers quietly refusing to grow a KV cache at all.

ตัวอย่าง

ลองเปิด Activity Monitor แล้วถาม Logy คำถามยาวๆ ดู เส้นกราฟหน่วยความจำที่แทบไม่ขยับนั่นแหละ คือ 30 ใน 40 เลเยอร์ของมัน กำลังปฏิเสธที่จะขยาย KV cache อยู่เงียบๆ

2026 · MarkTechPost, "Qwen team open-sources Qwen3.6-35B-A3B" — marktechpost.com · architecture detail per AICE Attention Canon §7; gating granularity and QK-Norm retention marked **[inferred]/[unconfirmed]** in the canon itself.

End of minibook. Source of truth: `~/Desktop/AICE/canon/ATTENTION_CANON.md`. Written 2026-07-31 for ICE.

จบเล่ม แหล่งข้อมูลหลักคือ `~/Desktop/AICE/canon/ATTENTION_CANON.md`
เขียนไว้เมื่อ 2026-07-31 สำหรับ ICE