Episode Book

EPISODE #6465

Why Modern AI Models Cut Most Attention Layers

PUBLISHED	5 November 2026 at 19:44 UTC
DURATION	25:56
LISTEN	myweirdprompts.com/episode/sparse-delta-rule-attention-hybrids-25441/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	z-ai/glm-5.3
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.26
SOURCES	25 cited by the research (listed at the end)
BOOK GENERATED	6 November 2026 at 11:12 UTC

SYNOPSIS

Standard softmax attention forces models to store every past key and value, making multi-turn chat memory balloon past model weights. Production models are quietly replacing full attention with layered hybrids instead of abandoning it completely. Gated delta rules compress history into fixed recurrent states, learned sparse indexers read only relevant tokens, and latent bottlenecks shrink what remains. These techniques stack together across open-weight architectures like Qwen and DeepSeek to slash inference costs and accelerate long-context decoding.

TRANSCRIPT



Corn

The line you hear everywhere is that the transformer won and attention is here to stay. The interesting news is the opposite: the transformer's core recipe is being taken apart in shipping models, not in papers. Daniel's prompt this week points straight at it. He's hearing a lot about Mamba, suspects it isn't the only alternative, and wants to know what else is different and actually reaching production — beyond Mamba and the hybrids built on it — and what's really at the forefront of the fundamental building block of inference today. His instinct is right, by the way. Mamba is one lineage among five, and it's not even the biggest one anymore.



Herman

It's a good moment for the question, too, because the answer changed shape this year. The argument stopped being attention versus no attention and became which layer does which job. That's the whole frame now.



Corn

Which is a strange thing to say about an architecture that was supposed to be indivisible. The original pitch of the transformer was "attention is all you need" — the whole point was that you throw out the recurrence and keep the attention.



Herman

Right, and the pitch held for seven years because the alternative kept losing. What changed isn't that someone found a better argument. What changed is that people found the exact seam where attention is wasteful, and started cutting there instead of trying to replace the whole thing at once. The revolution arrived as a series of budget cuts.



Corn

So let's start with the baseline everyone is trying to replace.



Herman

Herman Poppleberry here, and the baseline is this. Softmax attention keeps every key and value it has ever seen, so the KV cache grows linearly with context, and reading a text of n tokens costs you n squared. Concrete anchor, because that phrase is abstract until you



Herman (continued)

price it: Qwen3-32B, from April of last year, holds about thirty-two gigabytes of keys and values for a single 128K-token conversation. That is roughly half of what the model's actual weights take. The cache is half the model.



Corn

For one conversation.



Herman

For one conversation, yes. Serve a hundred of those and you're renting a small datacenter to store a chat log.



Corn

And it's worth being precise about why that hurts, because it's not just the memory bill. Every single token the model generates has to touch that whole cache. So the generation itself slows down as the conversation gets longer — the user's experience degrades in real time, not just the accounting.



Herman

The cost isn't static storage you pay once; it's a tax on every decode step. Which is why the fix everyone knows — Mamba, the state-space hybrids, we've done those episodes — is a fixed-size state. Compress or replace. But Daniel's asking what's beyond that, so let's be disciplined about it. What's the most widely deployed thing that isn't Mamba?



Herman

Gated delta-rule linear attention. And I want to unpack it, because the name makes it sound exotic and it's actually three ideas stacked up, each one simple. Start with linear attention, the Katharopoulos style: replace the softmax with a feature dot product, and attention turns into a recurrence with a fixed d-by-d state. Instead of keeping every key and value, you keep one matrix and update it as each token arrives. The cache stops growing entirely.



Corn

The library becomes one notebook.



Herman

One notebook page, exactly. Then the delta rule, which is Schlag, Irie and Schmidhuber — before you store a value under a key, you read what's already stored under that key and write only the difference. It's one gradient step on the memory's own error. If something's already there and it's wrong by a little, you correct it by a little instead of scribbling over it. Then gating on top of that — deciding how much of the old state to keep — gives you Gated DeltaNet, and that's the thing that actually shipped.



Corn

Where did it ship?



Herman

Qwen3-Next in September of last year, Qwen3.5 in February, Olmo Hybrid from AI2 in March. And Moonshot refined it into Kimi Delta Attention, with finer channel-wise gating, and built Kimi Linear on it — that one interleaves the delta-rule layers with full latent attention at three to one, cuts KV cache up to seventy-five percent, and at a million tokens it generates over six times faster than the equivalent latent-attention model.



Corn

Six times faster at a million tokens is the number that explains why everyone's doing this. But hold on — the notebook metaphor has a floor. A d-by-d state holds at most d key-value pairs exactly. What happens to pair number d plus one?



Herman

It blurs the ones already there. That's the trade, and the researchers are blunt about it — state size versus recall ability, you pick one. There's a lovely image from a writeup this year: the library has turned into one notebook page on which every note is written over the others. It stays readable while there are few notes.



Corn

So the fixed state is a photograph of everything, badly, rather than an album.



Herman

And here's the counterintuitive part. Someone ran the experiment on what happens when the page is full, and from sixty-four stored pairs onward, plain addition — just adding every new value in, no clever correction — recalls more than the delta rule does. Once memory is saturated, the delta rule keeps recent pairs sharp and old ones faint, while addition keeps a blur of everything in which the nearest value is still usually close enough to right.



Corn

The sophisticated method wins until the moment it loses, and then the dumb method wins by being uniformly mediocre.



Herman

That's the honest shape of it, yes. And it's why nobody ships a pure delta-rule model. Kimi K3, from July — 2.8 trillion parameters total, a hundred and four billion active, ninety-three layers — is sixty-nine delta-attention layers and only twenty-four latent-attention ones. Three of every four layers have the notebook. Every fourth layer gets the album.



Corn

Which is a weird thing to internalize. The frontier model is mostly not a transformer.



Herman

Mostly not, by layer count. It's a hybrid where the attention layers are load-bearing for exactly the thing the notebook can't do — exact recall.



Corn

And can you say anything about where in the stack those album layers sit? Is it every fourth layer like bricks, or is it clumped?



Herman

It's not uniform — the placement is a design decision each lab makes differently, and we'll come back to that with the ablation work later, because it turns out the placement matters more than anyone expected. Hold that thought too.



Corn

Two thoughts now. You're running a tab.



Herman

I always am. Okay. The delta rule trades recall for a fixed state. The other direction is to keep real attention and read less of it. That's the sparse stuff.



Herman

That's DeepSeek's line, and I think it's the most consequential of the five approaches for a boring reason: it's the default long-context mechanism in the Chinese open-weight frontier models now. Native Sparse Attention was the paper in February of last year; DeepSeek Sparse Attention shipped in V3.2-Exp in September. The mechanism is a lightning indexer — a cheap scorer that grades every past token, then full latent attention runs only over the top two thousand and change.



Corn

It grades every past token.



Herman

Every one. Keep that thought, it comes back with teeth later.



Corn

Noted. What does it buy?



Herman

At 131K context, DSA reads about 1.1 gigabytes per decode step, versus 5.2 for dense latent attention. Roughly five times less, and DeepSeek cut API prices by more than half the day it shipped. The speedup reports for the NSA formulation run up to eleven point six times on decoding at 64K context. And it's spread: Kimi K3.1 Pro gives each of its twenty-four attention layers an indexer with thirty-two heads, selecting the top two thousand forty-eight tokens per query no matter how long the context grows. Fixed cap. A million tokens of history, two thousand get read.



Corn

So the query says, of everything ever said, these two thousand sentences matter. It's a clerk with a good filing system.



Herman

And the filing system is learned, which is the difference between this and the hand-tuned sparse attention of a few years ago — sliding windows, strided blocks, things engineers dreamed up. The indexer is trained with the model. It learns what deserves reading.



Corn

Which is a little unnerving if you stop and think about it. There's no guarantee the clerk learned to file the things the benchmark cares about. If the indexer never learned that a specific serial number in paragraph four might matter later, no amount of downstream attention recovers it.



Herman

That's the failure mode, and it's why the retrieval benchmarks matter so much in these papers — you need a needle-in-a-haystack test to check whether the clerk is filing needles or just filing hay. The reported numbers hold up on those tests at ordinary context lengths. The interesting cracks appear at multi-million tokens, and we'll get there.



Corn

Underneath both of those, there's the quieter trick. Latent attention. Compress the cache itself.



Herman

Multi-head latent attention, from DeepSeek-V2, and it's the substrate everything else sits on now. The dense layers in Kimi K3 are gated latent attention; the sparse indexer grades latent-compressed keys. Mechanically it's a shared low-rank bottleneck — compress the key-value pairs into a smaller latent representation, decompress on the way out. Eighty-one percent KV cache reduction.



Corn

It's a filing cabinet where you photograph the documents in low resolution and shred the originals.



Herman

With the pleasant property that the compression is lossy in a principled way. There's a mechanistic study this year that trained a small transformer with the latent bottleneck and looked at what it actually keeps, and the answer is: content, not position. Entity identity survives at ninety-eight percent; positional information is discarded. Which validates the whole design — the rotary position encodings live outside the bottleneck precisely so the bottleneck is free to be a pure content store.



Corn

Ninety-eight percent. Somebody had to check that.



Herman

And I'm glad they did. Same study found the bottleneck is over-provisioned — it uses about forty-six percent of its capacity on average.



Corn

The compression could compress.



Herman

Which is the finding I'd file under good news. There's headroom. And the retrofitting work is the other thing worth flagging: there's a paper from October on converting existing multi-head and grouped-query models to latent attention, reporting up to five point three times serving throughput on a thirty-billion Qwen model. They call it MLAfication, which is a terrible name for a useful idea.



Corn

You don't convert the model's brain. You convert its storage.



Herman

And with four-bit KV quantization on top, the footprint drops by ninety-four-plus percent.



Corn

So — three of five covered, and I notice they compose. The delta rule kills the cache in most layers, latent attention shrinks it in the rest, the indexer cuts what gets read. None of these are rivals, exactly. They're a stack.



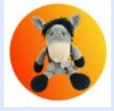
Herman

They're a stack, and that's where the field landed. But before we get to the synthesis, there are two more pieces, and one of them is the smallest change in the whole list.



Corn

The one-line change.



Herman

Gated softmax attention. You apply a head-specific sigmoid gate after the scaled dot-product — that's it. One modification, after the attention computation. The Qwen team tested thirty variants across a fifteen-billion mixture-of-experts model and a 1.7-billion dense one, three and a half trillion training tokens, and the finding is that this single gate consistently improves performance, stabilizes training, tolerates larger learning rates.



Corn

A gate after the multiplication. That's the sort of result that makes you suspicious.



Herman

It made me suspicious, and I went looking for why it isn't just noise. Two reasons the authors give. It introduces non-linearity into the low-rank mapping, and the gating scores are query-dependent, so each query effectively sparsifies which heads it listens to. And there's a specific bonus: it mitigates attention sink — that phenomenon where early tokens hoard attention mass for no semantic reason — and improves long-context extrapolation.



Corn

So the model stops treating the first sentence of the document as a load-bearing wall.



Herman

That's a fair gloss. And it's shipping. Qwen3.8-Flash-Next is explicitly a gated-delta plus gated-attention hybrid, and Kimi K3 lists gated attention output as a component. When a one-line change survives to frontier scale, it's real.



Corn

I'll take your word with a small reserve of doubt. What's the fifth?



Herman

The fifth is my favorite of the five, because it attacks the thing nobody was supposed to attack. Remember I said keep the thought about the indexer grading every past token?



Corn

It comes back with teeth, you said.



Herman

Zyphra said it plainly this week: although the final attention operation is sparse, the indexer itself must still search the entire context. This effectively moves the quadratic computation from attention into the indexer.



Corn

The fix for the quadratic cost is quadratic.



Herman

The fix for the quadratic cost is quadratic wearing a hat. And at very long context it takes over — Prefrontal AI put numbers on it last week with their Latent Sparse Attention: at ten million tokens, the indexer search eats ninety-eight percent of a sparse attention layer's time.



Corn

Ninety-eight. So your clever sparse layer is two percent attention and ninety-eight percent clerk.



Herman

The clerk became the department. Their answer, and Zyphra's Hierarchical Attention the day before, is a tree. Cluster the keys in a learned latent space, summarize chunks with learned reference queries, and route coarse to fine — you check the top of the tree, pick a branch, descend. Prefill drops from quadratic to T log T, and their headline is a seventy-two-times single-layer prefill speedup at ten million tokens.



Corn

Seventy-two is the number they'll print on the poster.



Herman

It is the number on the poster, and I'll do the poster's job for it: that's the best cell of the table. At a million tokens the single-layer gain is eight point two, not seventy-two, and the whole-model time-to-first-token gain is one and a half. Still real. Not seventy-two. And the accuracy cost is uneven — one model drops five points on a long-context benchmark, another trails its baseline by up to eight points on retrieval at multi-million tokens.



Corn

The blur creeps back in at the top of the tree.



Herman

Something creeps back in. Neither of these has shipped in a production model yet — code and kernels, research releases, no deployed weights I can point to as of this week.



Corn

Okay, before the synthesis — the three-into-one. The stack. You said that's the actual frontier.



Herman

It is, and there's a model that states the thesis out loud, which almost nobody does. GLM-5.3-Flash, from late August, from Z.ai — three hundred twenty billion total parameters, eighteen billion active, forty-five layers, a million-token context, MIT licensed. The design principle is explicit: linear attention layers carry local dependency modelling, sparse attention layers handle global retrieval. Each mechanism gets the job it's good at. Against the previous GLM it reports three times less attention compute and four point four times less KV cache.



Corn

And the pricing follows from the architecture, not the other way around.



Herman

Fifteen cents a million input tokens against a dollar forty for the predecessor. There's also a nice wrinkle — they pool four indexer key vectors into one to make the indexer cheaper, and the whole backbone is described in NVIDIA's docs as a hybrid delta-attention plus sparse-attention decoder. Forty-five layers where the previous model had ninety-two. Shallow and wide-jobs instead of deep and uniform.



Corn

And that layer-count drop is its own statement, isn't it. If ninety-two layers were necessary before, and forty-five do the job now, then a lot of those layers were doing redundant work that the specialized ones now split between them.



Herman

Or the redundancy was the price of uniformity. When every layer has to be able to do every job, you need more layers to cover the gaps. Specialization is what lets you halve the count without halving the capability. It's the same logic as any factory floor, honestly.



Hilbert

Medieval Pottage Facts. The hungry gap was the stretch of late spring when the winter stores ran out and the new crops had not yet ripened. People simply waited, and some of them did not.



Corn

A fixed pantry and an unripe field. The middle ages ran the same trade we're describing.



Herman

Grimly apt. And the modern version — Qwen4-Exp at the end of August was the first architecture integrating the linear and the sparse side in one release, and Kimi K3.1 Pro ships it too. The frontier is the combination. Three labs, same recipe.



Corn

Which brings me to the part you've been politely not saying. The counter-evidence.



Herman

MiniMax. Reported, not paper-confirmed, but consistent across sources: they went back to full attention in all sixty-two layers of M2 — a two-hundred-thirty-billion parameter model — after finding linear attention fragile on reasoning and multi-turn agentic tasks, even though it behaved fine on ordinary prompts.



Corn

Behaved fine until the agent needed to remember step forty.



Herman

And there's ablation work this year on Qwen3.5 and Falcon-H1 hybrids that found removing either pathway — the attention or the linear one — substantially degrades performance. Both contribute. The importance is position-dependent, concentrated in early or mid-network layers. So the hybrids aren't a free lunch; they're a lunch where nobody's sure who paid.



Corn

And that's the tab I was holding. The placement of the attention layers isn't decoration — the ablations say some layers can't be swapped and others barely matter, and right now nobody has a clean theory of which is which.



Herman

It's layer-by-layer empiricism. Which is uncomfortable for a field that likes its architectures to come with proofs.



Corn

That's the question underneath Daniel's, though, isn't it. Whether the division of labor is principled — linear for local, sparse for global, because that's what the math wants — or whether it's empirical tuning that happens to look principled, one architecture search away from being overturned.



Herman

And the agentic workloads are exactly the ones that stress it. An agent at turn two hundred needs exact recall of a tool call from turn three. That's what the fixed state blurs.



Corn

The workload driving long context is the workload the fixed state handles worst.



Herman

Well put in the sense that I was going to say it and you said it faster. There's one more wrinkle I'd be dishonest to leave out, and it's the dark horse. A head-to-head this year compared xLSTM, Mamba-2 and Gated DeltaNet on code and time series, and the xLSTM family beat the others in nearly all comparisons — better state tracking, better counting-like accumulation. And there's a separate claim that xLSTM Pareto-dominates transformers outright. It's the backbone of TiRex, one of the best time-series foundation models, and it gets a fraction of Mamba's mindshare.



Corn

The oldest idea in the room — it's a gated recurrent unit's grandchild — and it's quietly winning the benchmarks nobody tweets about.



Herman

Which is a good note to leave the mechanisms on. Five approaches, one stack, one counter-narrative, one dark horse.



Corn

I was going to ask which one you'd bet on.



Herman

I'd bet on the stack, not any single piece of it. And I'd bet against anyone who says the current ratios are settled.



Corn

You said the same thing about a different stack once and it held up fine.



Herman

The other thing I keep noticing — the indexer problem is the same shape as the cache problem was two years ago. Each fix relocates the cost and then becomes the thing needing a fix. Someone will be attacking the tree's root layer next year, and it'll be the same argument one level up.



Corn

It's moving expenses all the way down.



Herman

And the labs that win are the ones who notice the moving truck before it's loaded.



Hilbert

The indexer is the whole bill, and I've been saying that since before it was a tree. I ran one of these on a rented card for a month. H200, back in the spring, sixty hours I paid for out of my own pocket, and I kept a page per model — which layers were delta-rule, which were attention, written out by hand, because I don't trust a config file to still say the same thing on Thursday that it said on Monday. I've got eleven pages. Kimi K3 alone took most of a Sunday.



Corn

Eleven pages. You transcribed the layer stack of every hybrid you could reach.



Hilbert

The ones worth reaching. The bill was the surprise. They advertise the cache savings and they're real, right up until the context passes a certain length, and then your money goes to the thing they didn't put in the headline. Herman, you gave the numbers for it earlier, took you about forty minutes to get there, but you got there.



Herman

Forty minutes is about my average.



Hilbert

The card went back, the pages stayed. I know exactly which page has the model that stalled. It's in the drawer with the good pen.



Corn

You're not lending the pen.



Hilbert

The pen stays where it is. Whoever writes those kernels works harder than anyone thanks them for, and I'll hear nothing against them.



Corn

He's right about the bill, though. The hidden line item is the line item.



Herman

It usually is. And his eleven pages are more documentation than some of these labs publish.



Corn

That's the open question I'd leave with Daniel's phrasing — what's really at the forefront. As of this week it's the hybrid stack, linear for local, sparse for global, and nobody knows if the ratios are physics or fashion. MiniMax's retreat says fashion is on the table. If a second frontier lab walks linear attention back, the whole division of labor gets renegotiated. And the indexer's own cost is next in line for dismantling — the tree people already have their knives out. The transformer's basic formula stopped being the basic formula in production a while ago. What replaces it is still being argued with layer counts.



Herman

The way to watch it is the ablations, not the launches. When the next paper says which layers can't be swapped, that's the map of what's load-bearing.



Corn

Producer credit to Hilbert Flumingtop, as always, at the desk and on the pen. If this was your kind of episode, go back for episode three, Safetensors or something else; episode eighteen, Beyond the GPU; and episode twelve, The AI Breakthrough. This has been My Weird Prompts, the human-AI collaboration podcast. If today's tour earned its thirty minutes, the review box is right there — a line about your favorite layer type helps other listeners find us. We'll be back soon.



Herman

See you then.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **model card, 321.3B params, MIT, created 2026-08-25, 6,277,088 downloads/30d; retrieved 2026-11-05. [primary]**
   https://huggingface.co/zai-org/GLM-5.3-Flash

2. **Zyphra, ZHA: Zyphra Hierarchical Attention, 2026-11-02; retrieved 2026-11-05. [primary]**
   https://www.zyphra.com/our-work/zyphra-hierarchical-attention

3. **DeepSeek, Introducing DeepSeek-V3.2-Exp, 2025-09-29; retrieved 2026-11-05. [primary]**
https://www.deepseek.com/en/news/v3-2-exp/

4. **Native Sparse Attention: Hardware-Aligned and Natively Trainable Sparse Attention (NSA); retrieved 2026-11-05. [primary]**
https://arxiv.org/pdf/2502.11089

5. **vLLM blog, DeepSeek-V3.2-Exp in vLLM: Fine-Grained Sparse Attention in Action, 2025-09-29; retrieved 2026-11-05. [primary]**
https://vllm.ai/blog/2025-09-29-deepseek-v3-2

6. **NVlabs repo, integration dates 2025-09-10 / 2026-02-17 / 2026-03-12; retrieved 2026-11-05. [primary]**
https://github.com/nvlabs/GatedDeltaNet

7. **Kimi Linear model card, 2025-10-30; retrieved 2026-11-05. [primary]**
https://huggingface.co/moonshotai/Kimi-Linear-48B-A3B-Instruct

8. **repo, created 2025-10-29, MIT; retrieved 2026-11-05. [primary]**
https://github.com/MoonshotAI/Kimi-Linear

9. **vLLM PR #27809, merged 2025-10-30; retrieved 2026-11-05. [primary]**
https://github.com/vllm-project/vllm/pull/27809/files

10. **Kimi K3 model card, 2.8T/104B, 93 layers, 69 KDA + 24 Gated MLA, July 2026; retrieved 2026-11-05. [primary]**
https://huggingface.co/moonshotai/Kimi-K3

11. **Kimi K3: Open Frontier Intelligence technical report; retrieved 2026-11-05. [primary]**
https://arxiv.org/html/2607.24653v2

12. **Gated Attention for Large Language Models: Non-linearity, Sparsity, and Attention-Sink-Free, Qiu et al.; retrieved 2026-11-05. [primary]**
https://arxiv.org/abs/2505.06708v1

13. **Eva Koroleva, Hybrid attention: why new LLMs keep full attention in only a few layers, 2026; retrieved 2026-11-05.**
https://xmarva.github.io/blog/2026/hybrid-attention/

14. **On Subquadratic Architectures: From Applications to Principles, arXiv; retrieved 2026-11-05.**
https://arxiv.org/html/2606.12364

15. **The Evolution of Attention in Large Language Models: Mechanisms, Trade-offs, and Emerging Trends; retrieved 2026-11-05.**
https://arxiv.org/abs/2609.39661

16. **AI2Work, GLM-5.3-Flash Mixes Sparse and Linear Attention in Open Weights, 2026-08-26; retrieved 2026-11-05.**
https://ai2.work/blog/glm-5-3-flash-mixes-sparse-and-linear-attention-in-open-weights

17. **sysnote, Prefrontal's Latent Sparse Attention speeds up one attention layer 72x at 10M tokens, 2026-11-03; retrieved 2026-11-05.**
https://sysnote.net/news/prefrontal-s-latent-sparse-attention-speeds-up-one-attention-layer-72x-at-10m-to

18. **DeepSeek Sparse Attention from First Principles; retrieved 2026-11-05.**
https://www.tensoreconomics.com/p/deepseek-sparse-attention-from-first

19. **TechCrunch, 2025-09-29; retrieved 2026-11-05.**
https://techcrunch.com/2025/09/29/deepseek-releases-sparse-attention-model-that-cuts-api-costs-in-half/

20. **AMD, Kimi K3.1 Pro on AMD Instinct GPUs, Nov 2026; retrieved 2026-11-05.**
https://www.amd.com/en/developer/resources/technical-articles/2026/kimi-k3-1-pro-on-amd-instinct-gpus.html

21. **OpenModelAtlas, Kimi K3 layer breakdown; retrieved 2026-11-05.**
https://atlas.klimoza.dev/models/kimi-k3/blocks

22. **MindPattern, transformers v5.16.0 adds Qwen4-Exp, 2026-08-27; retrieved 2026-11-05.**
https://mindpattern.ai/s/2026-08-27-transformers-v5-16-0-adds-qwen4-exp-the-first-hybrid-combining-linear-and-sparse-att
e

23. **Haller, Golde & Akbik, What Matters in Linearizing Language Models?, EMNLP 2026, pp. 60257**
60274; retrieved 2026-11-05.
https://aclanthology.org/2026.emnlp-main.2648.pdf

24. **Component Ablation for Efficient Hybrid Language Model Architectures; retrieved 2026-11-05.**
https://arxiv.org/abs/2603.22473v2

25. **Through the Bottleneck: How Multi-head Latent Attention Separates Content from Position; retrieved 2026-11-05.**
https://arxiv.org/abs/2607.23054v1