Episode Book

EPISODE #6220

Mixture of Experts: Bigger Than It Looks

PUBLISHED	23 October 2026 at 09:19 UTC
DURATION	20:10
LISTEN	myweirdprompts.com/episode/mixture-of-experts-active-parameters-33076/
HOSTS	Corn, Herman, Daniel, Hilbert
SCRIPT MODEL	meta/muse-spark-1.3
VOICES	chatterbox-regular+meorav (chatterbox-local)
PIPELINE	V5.20
SOURCES	13 cited by the research (listed at the end)
BOOK GENERATED	24 October 2026 at 07:26 UTC

SYNOPSIS

Mixture of experts is sold as the trick that makes giant models cheap. It doesn't — it makes them bigger. This episode unpacks what the active parameter count actually measures, why a trillion-parameter checkpoint can run like a 3.6B model, and why the router fires a committee of experts per token instead of one specialist. We walk through the design paper arguing you should maximize total parameters and top-k rather than shrink active, the granularity and width-to-depth sweet spots, and the load-balancing loss that keeps 32 experts from collapsing into three. Then the harder flip: the March paper arguing MoE pays a double inference penalty, with DeepSeek V3 losing roughly 4.5x throughput to a quality-matched dense baseline at long context because expert weights crowd out KV cache. Store an elephant, ride a bicycle.

TRANSCRIPT



Corn

Everyone keeps saying mixture of experts makes models cheaper. It does not. It makes them bigger.



Herman

That is the part the model cards hide in plain sight. You see a trillion parameters and think supercomputer, then the active number is tiny by comparison.



Corn

Daniel is in the studio today, and he brought the exact confusion with him.



Daniel

Yeah, I keep seeing these active counts all over Hugging Face and my gut says smaller active must mean better. Fewer weights firing, more selective routing. Is that roughly right.



Corn

The short version is you noticed open models crossing a trillion parameters, you noticed active counts bouncing all over the place, and you want to know what the number measures, why tuning it separates models, and what you would actually feel between a wide active slice and a narrow one in chat or prediction.



Herman

And that gut feeling about smaller being better, Corn, that is where we should start, because it is half right in a way that makes it wrong.



Daniel

Okay, so what does the active number even count.



Herman

Take a normal transformer block. Usually there is one big feed forward network after attention. In mixture of experts you rip that out and replace it with a bank of smaller feed forwards, call them experts, plus a little learned router. Every token gets scored, top k experts win, their outputs get mixed by the router weights.



Corn

A trillion parameters doing shift work.



Herman

Total parameters count every expert sitting in the checkpoint. Active counts only the ones that fire for a given token, plus the stuff that is always on. Attention, embeddings, any shared experts that run every time.



Daniel

Per token, not per prompt.



Herman

Per token, yes. Two words in the same sentence can take completely different paths. The Hugging Face line on this is clean. Model capacity depends on total parameters, but inference speed depends on active parameters.



Daniel

But how does that work in practice? Does the router look at the word itself, or the whole context around it?



Herman

It looks at the hidden state for that token at that layer, so it has already absorbed context from attention. It is a tiny linear layer that spits out scores for each expert, then you take the top k and softmax just those. So "bank" in "river bank" and "bank" in "savings bank" can score completely different experts by layer six or seven.



Corn

Same word, different plumbers.



Daniel

And that scoring happens at every MoE layer, for every token?



Herman

Every MoE layer, every token, every forward pass. That is why routing quality compounds. A small mistake early reroutes everything downstream.



Corn

Capacity in storage, speed at the till.



Herman

Worked example helps. Take gpt oss twenty b. Twenty one billion total in the checkpoint. Thirty two experts in each MoE layer, four active per token. So per token you only pay for about three point six billion active.



Daniel

So it stores like a twenty billion model and runs like a three and a half billion model.



Herman

In compute per token, roughly, yes. Memory is the catch, and we will get there, because you still have to store all twenty one billion somewhere. But the arithmetic per token is that small.



Corn

Store an elephant, ride a bicycle.



Daniel

What do you think, Corn, does that match how you picture it.



Corn

I picture a hotel where only four rooms ever have the lights on, but you still pay to heat the building. Which is why your smaller is better idea feels tidy and is probably doomed.



Herman

It is doomed in an interesting way. Sparsity does push experts to specialize. When routing gets sparser, expert neurons get less polysemantic, less doing five jobs at once, compared to a dense feed forward. So your instinct about selectivity is not crazy.



Daniel

But what about the failure mode? If only four out of thirty two ever fire, do the other twenty eight just atrophy?



Herman

They can, and that is called expert collapse or load imbalance. Without an auxiliary load-balancing loss, the router finds two or three experts it likes and hammers them, while the rest starve. So training has to actively force traffic to spread out, otherwise your thirty two experts are really three experts and twenty nine tourists.



Corn

Tenured tourists.



Daniel

But they are not specialists the way I imagined, right. I imagined a biology expert, a law expert.



Herman

No, and that is the fun part. They turn out to be fine grained task experts. Linguistic operations, little semantic jobs. The canonical example going around this year is an expert that basically does closing brackets in LaTeX.



Corn

Ten years of training to shut parentheses.



Herman

Not biology versus law. More like this cluster of neurons learned when a curly brace needs to die. That is specialization, just at a level humans find anticlimactic.



Daniel

Is that really all they do? Just punctuation?



Herman

It is more like micro-syntax plus context. One expert might handle closing brackets, but only when they close a math environment, not plain text. Another might handle past-tense verbs after proper nouns. Humans want chapters. The model learns commas.



Corn

I spent English class the same way. Learned nothing but commas.



Daniel

So sparser gives more of that narrowness, which sounds like an argument for smaller active.



Herman

Here is where it flips. There is a design paper from January that tried to put this on rails. Their claim is total and active alone are not enough to describe an optimal architecture. Performance is mostly set by total parameters and sparsity, where sparsity is number of experts divided by top k.



Corn

Such a dry ratio for such a weird idea.



Herman

And their recommendation is the opposite of minimize active. Maximize total under your memory limit, minimize sparsity, which means maximize top k, and minimize number of experts. In plain English, fire more experts per token, keep fewer of them sitting around.



Daniel

Wait, fire more. That is literally backwards from my guess.



Herman

What do you make of that, Corn. Does bigger top k feel wasteful to you.



Corn

It feels like hiring four plumbers to watch one sink. But if each one only knows lefty loosey, maybe you need the crowd.



Herman

That is actually close. More active experts means more capacity engaged per token, and the router gets far more combinations to play with. Selectivity is about routing quality, not about how few you fire. Two experts can be badly chosen. Eight can be well chosen.



Daniel

So how many combinations are we actually talking about? With eight out of a hundred and twenty eight, that has to be enormous.



Herman

It is combinatorial. Eight out of a hundred and twenty eight is more than thirty billion possible sets per layer, and you have dozens of layers. Even four out of thirty two is thirty six thousand combos per layer. The router is not picking an expert, it is picking a committee, and a slightly different committee per token.



Corn

Congress, but it works.



Daniel

So fewer is not more selective, fewer is just fewer.



Herman

Yes. And there are two other knobs that matter as much as the headline numbers. Granularity, which is model width divided by expert width, sweet spot around four to eight, diminishing returns past that. And width to depth ratio, sweet spot around forty two, keep it between thirty two and sixty four.



Corn

Forty two. Of course.



Herman

I know, I laughed too. But even with total and active fixed, you can build many different models out of depth, width, expert count, active count, granularity. That is why tuning it is a differentiator. The card shows two numbers. The design hides five decisions.



Daniel

Can you make granularity concrete? If I fix active at, say, twenty billion, what changes if I split it into tiny experts versus chunky ones?



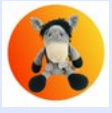
Herman

Think pizza. Same total dough, you can cut eight big slices or sixty four tiny squares. Tiny squares let you assemble a very precise plate for each token, a little syntax here, a little factual recall there. But go too tiny and each expert is too weak to learn anything coherent, and routing overhead explodes because you are scoring hundreds of experts per token.



Corn

Death by tapas.



Herman

Four to eight hits the balance. Big enough to be useful, small enough to mix.



Daniel

And DeepSeekMoE was the recipe for that.



Herman

That was the formalization. Split experts into smaller ones so you can mix and match more flexibly, and peel off a few shared experts that always run to soak up common knowledge. Fine grained segmentation plus shared expert isolation. That cut redundancy and let the routed experts actually diverge.



Corn

We did the fine tuning episode years back, the one about how a tiny tune changes which weights matter. This is the cousin. Tuning changes what matters, routing changes what wakes up.



Herman

That is a good way to put it. Same moral. The checkpoint is not the behavior.



Daniel

Give me a concrete win for that recipe, not just theory.



Herman

DeepSeekMoE at two billion parameters matched GShard at two point nine billion, so about one and a half times the expert compute, and nearly matched a dense model of the same total size. Smaller active budget, same quality, because the experts stopped tripping over each other.



Daniel

What were they tripping over before? Just learning the same thing twice?



Herman

Mostly common knowledge and stopwords. Every expert was relearning "the" and "and" and basic grammar, wasting capacity. Once you give that to a shared expert that is always on, the routed experts are free to diverge into those weird bracket-closers and tense-handlers. Redundancy drops, diversity goes up.



Corn

Less overlap, more closing brackets.



Herman

Do you buy that tradeoff, Corn, or do you still want the tiny active number for elegance.



Corn

I want the tiny number the way I want a tiny electric bill. Whether it writes better is a separate question, Herman.



Daniel

Okay, that is design. What about deployment. If active sets speed, MoE should be faster, right.



Herman

That is the second flip, and it is harsher. There is a March paper with the qs inequality. Their argument is MoE pays a double penalty at inference. Routing fragments your microbatch, so different tokens want different experts and you lose weight reuse. And the whole resident expert pool sits in high bandwidth memory that could otherwise hold KV cache.



Corn

A hotel lobby full of luggage, no room for guests.



Herman

For DeepSeek V three at a hundred and twenty eight k context, they say a quality matched dense baseline gets about a four and a half times throughput advantage. Four and a half times.



Daniel

That is huge. So MoE is not an inference win at all.



Herman

Their line is training time flop efficiency is a poor proxy for inference time economics. MoE may be best viewed as a training time optimization, with distillation into dense models as the path to inference efficient deployment.



Daniel

Wait, why does the KV cache matter so much here? I thought that was separate from weights.



Herman

It is separate, but they share the same HBM. At long context, KV cache balloons. A dense model uses its HBM mostly for cache, so it can serve huge batches. An MoE has to keep hundreds of billions of expert weights resident, even the idle ones, so there is far less room left for cache. Smaller batches, lower throughput, higher cost per token.



Corn

Train fat, serve thin.



Daniel

Corn, would you rather have the giant MoE or the distilled dense if you are just chatting.



Corn

I would rather have whatever answers before I fall asleep. Which, given my pace, is a low bar, Herman, and most models still miss it.



Herman

You would feel it in two places. Larger active means more compute per token, higher cost and latency, but more capacity engaged per token. Smaller active means cheaper and faster for the same stored capacity, but you lean harder on routing quality. Bad routing with narrow active hurts more.



Daniel

Is there a fun parallel here? This feels like the brain thing where we only use a fraction at a time.



Herman

It is the analogy everyone reaches for, and it is half useful. Your brain has about eighty six billion neurons and only a sparse subset fires for any task. But neurons are slow and massively parallel with near-zero movement cost. GPUs are the opposite, fast and starved for memory bandwidth. So biology gets sparsity almost for free, we pay for it in routing and HBM.



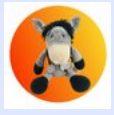
Corn

My brain fires a sparse subset too. Usually the wrong subset.



Daniel

Can we ground that in real cards.



Herman

Let us do it. DeepSeek V three, six hundred seventy one billion total, thirty seven billion active. On Hugging Face it lists as six hundred eighty five billion because it includes a fourteen billion multi token prediction module. Trained on about fourteen point eight trillion tokens, about two point seven eight eight million H eight hundred hours.



Corn

A rounding error in GPUs.



Herman

Qwen three two thirty five B A twenty two B, two hundred thirty five billion total, twenty two billion active, a hundred and twenty eight experts with eight active, ninety four layers. Little sibling Qwen three thirty B A three B, thirty point five billion total, three point three billion active, same a hundred and twenty eight experts and eight active, forty eight layers.



Daniel

Same router shape, very different bodies.



Herman

Mixtral eight by seven B, the early one, forty six point seven billion total, about twelve point nine billion active, eight experts top two. And Kimi K two Instruct, one thousand twenty six point four billion total. That is your trillion parameter open model.



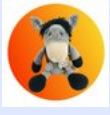
Corn

A trillion parameters and it still cannot load the dishwasher.



Daniel

For chat, what is the felt difference between the thirty B A three B and the two thirty five B A twenty two B.



Herman

Qwen markets the small one as efficient general purpose dialogue, the big one adds reasoning depth. In practice the small active can run in tight places, the Hugging Face blog notes gpt oss twenty b running on an M three Ultra Mac, while the large active wants datacenter hardware, more memory bandwidth, longer time to first token under load, higher per token price.



Daniel

So in a long agentic run, the small one would stay coherent but lose the thread on multi-step math, while the big one holds the plan but costs a fortune?



Herman

That is the typical report. Narrow active is fine for snappy back-and-forth, summarization, single-step tools. Wide active holds up better when you need five hops of reasoning without dropping state, because more capacity is engaged on each token to keep the plan alive. You pay for that memory in latency and dollars.



Daniel

So narrow active is local and snappy, wide active is smarter but heavier.



Herman

Often, but the card number lies by omission. Two models can share the same total and active and differ entirely in granularity, depth to width, shared expert count. That is why active alone does not tell you which is better for your task.



Corn

Same weight, different animal.



Herman

Which part worries you more as someone shopping for open weights, Daniel, routing quality or memory pressure.



Daniel

Memory, honestly. I shop for open weights and I care if it fits and if agents stay reliable. A narrow model that fits but routes badly sounds worse than a slightly wider one I have to host elsewhere.



Herman

That matches the literature. Narrow leans on the router. If the router misfires, those three billion active are the wrong three billion.



Corn

Wrong plumbers, flooded sink.



Hilbert

Activated is the wrong word. It is borrowed from bloody physiology, nerves firing, and it makes people picture switches clicking. Nothing clicks. It is arithmetic.



Herman

Okay, what would you call it, Hilbert.



Hilbert

Engaged. You engage four men out of thirty two. I had thirteen routing boards in the mid eighties, telephone boards, not computers. Copper and jacks. I set the overflow to spill to specialists when the main desk got busy.



Corn

Thirteen boards. Naturally.



Hilbert

The specialists were my cousins. All eleven of them. They were meant to take directory inquiries. They used the lines to run a tip sheet on greyhounds. Kempton Park, every afternoon, two forty five. Cost me forty seven pounds in one quarter in nineteen eighty five, trunk calls. I have not used the word activated since.



Herman

That is uncomfortably close to the LaTeX bracket finding. Your experts specialized, just not in the assigned domain.



Hilbert

They specialized in the three thirty at Kempton, Herman. You took twenty minutes to say experts learn narrow jobs. My cousins learned narrow jobs in a week.



Corn

Did the router learn, or did you just unplug Kempton.



Hilbert

I unplugged nothing. I left them to it and hired a girl who actually answered. Still paid the forty seven pounds. Still have the bill spike memorized, second quarter, eighty five. Your sparsity ratio does not cover cousins.



Daniel

So narrow routing only works if you check what they actually do.



Hilbert

Check every Friday. That is the whole job.



Herman

He is right about the checking, even if the cousins are doing crimes. Narrow active lives or dies on whether anyone audits the router.



Corn

Audit your cousins. Words to live by, Herman.



Herman

If MoE is mostly a training time win and the efficient deployment is a distilled dense model, what does that mean for this rush to ever larger total counts in open source.



Corn

It means the card is a menu, not the meal. Active tells you the price per bite. It does not tell you who cooked, how many cousins in the kitchen, or whether it fits on your Mac.



Daniel

That lands. I came in thinking smaller active meant smarter routing. I am leaving thinking active is just the slice you pay for per token, and the real work is everything behind it.



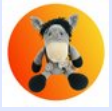
Herman

And the slice still matters for you. If you want local chat, chase small active with good routing. If you want depth and you have the hardware, pay for wide.



Corn

Thanks to Hilbert Flumingtop for producing the show.



Herman

For more along these lines, there's episode three, Safetensors or something else; episode eleven, How Does Fine Tuning Work Anyway; and episode five, Fine-Tuning ASR For Maximal Usability. This has been My Weird Prompts, the human AI collaboration podcast.



Corn

If you liked this one, leave us a review. It helps more than you think.



Daniel

See you tomorrow, and I will audit my routers.



Corn

We will be back soon.

SOURCES

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

1. **DeepSeek-V3 model card , created 2024-12-25 [primary]**
   https://huggingface.co/deepseek-ai/DeepSeek-V3
2. **Qwen3-235B-A22B model card , created 2025-04-27 [primary]**
   https://huggingface.co/Qwen/Qwen3-235B-A22B
3. **Qwen3-30B-A3B model card , created 2025-04-27 [primary]**
   https://huggingface.co/Qwen/Qwen3-30B-A3B
4. **Mixtral-8x7B model card , created 2023-12-10 [primary]**
   https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1
5. **Kimi-K2-Instruct model card , created 2025-07-11 [primary]**
   https://huggingface.co/moonshotai/Kimi-K2-Instruct
6. **Towards Principled Design of Mixture-of-Experts Language Models under Memory and Inference Constraints, 2026-01-13**
   https://arxiv.org/abs/2601.08215
7. **The qs Inequality: Quantifying the Double Penalty of Mixture-of-Experts at Inference, 2026-03-09**
   https://arxiv.org/abs/2603.08960
8. **The Expert Strikes Back: Interpreting MoE Language Models at Expert Level, ICML 2026**
   https://arxiv.org/abs/2604.02178
9. **DeepSeekMoE: Towards Ultimate Expert Specialization, 2024-01-11**
   https://arxiv.org/abs/2401.06066
10. **Mixture of Experts (MoEs) in Transformers, Hugging Face**
    https://huggingface.co/blog/moe-transformers

11. **Raschka FAQ, Why MoE models have fewer active parameters**
https://sebastianraschka.com/faq/docs/why-moe-huge-params-lower-active-compute.html

12. **NVIDIA Technical Blog**
https://developer.nvidia.com/blog/dense-vs-moe-models-active-parameters-throughput-and-when-to-choose-each/

13. **Wikipedia, last edited 2026-10-22 (Mistral Large 3: 675B total / 41B active)**
https://en.wikipedia.org/wiki/Mistral_AI