Keiji AI

TUTORIAL · OPEN-WEIGHT MODEL TRAINING AND EVALUATION

# How LLM Training Actually Works

## You don’t need the biggest model. You need the right data.

A practical guide for domains where expertise matters, quality can be verified, and your own data can become a compounding advantage.

Keiji AI Inc · keiji.ai · jimeng@keiji.ai

Jimeng Sun · August 15, 2026

TL;DR

# The whole document in one page

Nine claims the rest of this tutorial defends — starting from the two case studies in Part 0, where small specialized models beat the frontier outright.

01 **Two numbers decide everything.** Total parameters set memory and hardware cost; activated parameters set speed. In every current frontier model they are different numbers, and a vendor quoting only one is hoping you don't ask.

02 **All knowledge arrives in pre-training.** Fine-tuning transfers form reliably and facts badly — training on knowledge the base model lacks raises the hallucination rate. Tune for form; retrieve for facts.

03 **Reinforcement learning only works where answers are verifiable.** That is why every current model is strong at math and code and weaker at open-ended judgment. "Strong at agentic coding" does not transfer to "reliable at adverse-event extraction."

04 **Distillation is the practical route to a model you can host.** The teacher can be a stronger model, human experts, or both. Behavior cloning is the baseline; the expense is producing representative demonstrations, not the training run.

05 **Distillation raises floors, not ceilings.** Students imitate style before capability, and human preference evaluation hides the difference. If your evidence is "our reviewers liked the outputs," you have measured nothing.

06 **Open weights and runnable have become different properties.** Kimi K3 is genuinely open and needs roughly eighteen H100s just to hold its weights. GLM-5 and DeepSeek-V4-Flash matter more to you than the headline leader.

07 **Do not self-host first.** Evaluate on the labs' own APIs, move to a dedicated serving provider, self-host only when volume, residency or validation demands it. Pin the provider and the quantization or you cannot reproduce a result.

08 **Licence is a gating criterion, and it follows the checkpoint, not the brand.** Read the LICENSE file in the weights repository before the technical evaluation. A fine-tune inherits its base model's terms.

09 **The weights depreciate; the evaluation appreciates.** Both cases in Part 0 ran the same order of operations — the labeled data first, the model second. Spend the year on adjudicated gold standards and a verification harness, not on a domain model you will rebuild anyway.

READ ANY MODEL RELEASE BY ANSWERING THREE QUESTIONS

**Can we run it?**
Total parameters × bytes, plus KV cache. Ask what precision the quote assumes.

**May we use it?**
Revenue triggers, attribution at scale, field-of-use and jurisdictional limits.

**Is it good enough?**
On *your* task. A headline benchmark on software engineering tells you nothing.

If you are deciding whether your organization should run its own models, you will be handed numbers like “2.8 trillion parameters,” “104B activated,” “1M context,” “MoE,” “on-policy distillation,” “MXFP4.” Vendors will use these numbers to mean whatever they need them to mean.

This document explains what each of them actually is, using three real models released in the last six months as the worked examples. By the end you should be able to read a model release and answer three questions on your own: **can we run this, can we legally use it, and will it be good enough for our task.**

No mathematics. Some of the analogies are lossy — the ones that break are flagged. Part 0 starts with two published cases where a small specialized model beat the frontier outright, because that is the reason any of the rest is worth your time.

## Part 0 · Two cases where the small model won

Most AI budgets are built on a surprisingly passive assumption: that capability belongs to whoever has the biggest model, and everyone else should wait for the frontier to advance and then integrate it. The evidence is starting to point in the opposite direction. Two published results from the past year found that smaller models, trained on data the organization already owned, outperformed every frontier model they were tested against — while costing roughly an order of magnitude less to run. Neither breakthrough required a new algorithm. The advantage came from the data.

That changes the strategic question. The winners may not be the organizations that wait longest for better models. They may be the ones that realize the most valuable AI asset is already inside their walls — and start compounding it now.

### Case 1 · Financial judgment — an open-weight student beats the frontier

In June 2026 Bridgewater’s AIA Labs, working with Thinking Machines Lab, published results on six information-filtering jobs drawn from investors’ daily work: is this article relevant to a macro investor, does this central bank document signal a rate change, where does the boilerplate in this email begin. Trivial for their analysts — and the analysts could not articulate the rule they were applying.

Frontier models — Gemini, Claude and GPT variants — averaged roughly 50% accuracy on a plain statement of each task: a coin flip. Expert-written prompts and task reframing lifted them to the mid-70s, and then progress stopped; automatic prompt optimization added nothing, and the best model tested reached 78.2%, below the 80% their investors would trust. Newer releases were barely better per dollar. A fine-tuned open-weight model reached **84.7%** — **29.8% fewer mistakes than the best frontier model**, at **1/13.8th the inference cost per task**.

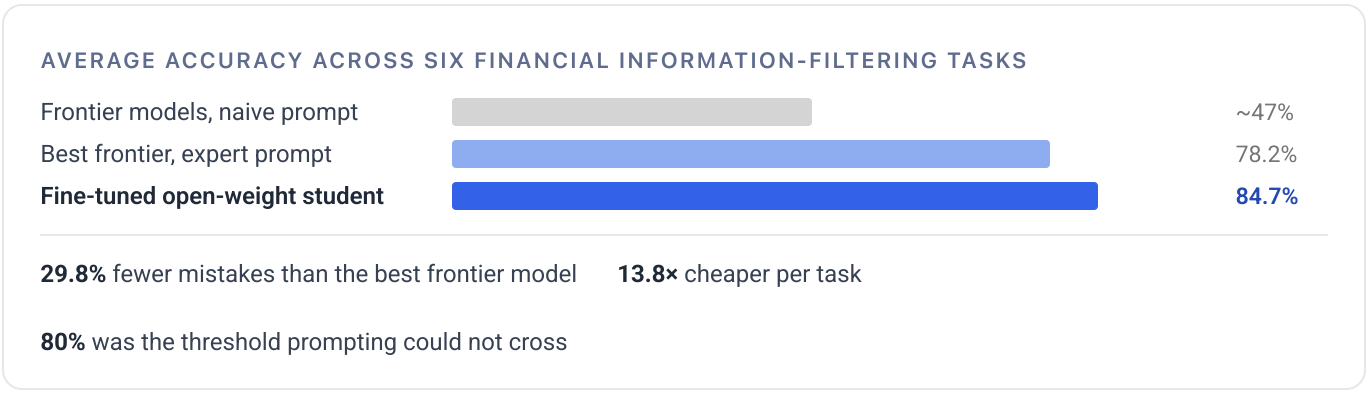



**Figure 1 —** Bridgewater AIA Labs with Thinking Machines Lab, June 2026. Prompting plateaus below the usable threshold; expert-labeled data crosses it.

The decisive input was not the training recipe — it was label quality, and the way they discovered it is the transferable part. Their first training set came from commercial vendors using **non-expert labelers**, and models trained on it stayed poor: not because the training was wrong but because **the labels were wrong**. Rather than pay experts to relabel everything, they trained on the cheap labels and then routed only the examples where the model disagreed with its own training label to their investors. If a model cannot reproduce an example from its own training set, either the case is genuinely hard or the label is bad. Expert attention went only to the contested cases.

Two details generalize beyond finance. Their tasks were ones where experts *have* judgment but cannot state the rule — exactly the shape of adverse-event triage, eligibility screening and endpoint adjudication. And an explicit prompt can only carry the part of an expert's intuition they can put into words; training is how you transfer the rest. The mechanics they used are the subject of Part 3.

### Case 2 · Clinical trial reasoning — 8B beats 70B on all eight tasks

The second case is our own, and it followed the same order of operations. We first built **TrialPanorama** — 1.6 million trial records from 15 registries, 152,000 samples across eight tasks spanning systematic review (search, screening, evidence summarization) and trial design (arm design, eligibility, endpoints, sample size, completion). Only then did we train against it: an **8B model**, post-trained with supervised fine-tuning and reinforcement learning, **outperforms 70B generic models across all eight tasks** (npj Digital Medicine, 2026). Roughly a ninefold reduction in activated parameters, against a better result, on clinical reasoning specifically.

SPECIALIZED, TRAINED ON TRIALPANORAMA

**8B**

~4.5 GB of weights at 4-bit · single workstation GPU · comfortably over 100 tokens/second

**Wins 8 of 8 tasks**

GENERIC, LARGER

**70B**

~35 GB at 4-bit · multi-GPU · and a 2.8T frontier model is a rack, not a server closet

**Loses 8 of 8 tasks**

**Figure 2 —** TrialPanorama, npj Digital Medicine (2026). The route to a deployable clinical model was not waiting for frontier models to shrink — it was building the benchmark, then training against it.

### What the two cases share

**The work was judgment-heavy and unwritten**
Experts performed it easily and could not specify it. That gap is where a prompt fails and training succeeds.

**The data came first, the model second**
Expert labels in one case, a published benchmark in the other. In neither was the model the asset.

**The winner was small enough to host**
Both students run on hardware you can own, inside your data boundary, with pinnable weights.

**Frontier progress did not close the gap**
Successive frontier releases improved little on these tasks, and cost more. Waiting was not a strategy.

> The pattern is what the finance team called *differentiated intelligence*: models tuned to one organization's needs outperforming general ones on that organization's work. **The frontier model was never the asset — the labeled judgment was.** The rest of this document is the working knowledge you need to act on that: which training stage does what, what you can change about a model you download, what it costs to run, and how to tell whether it worked.

## Part 1 · The two numbers that decide everything

Start here, because almost every confused procurement conversation traces back to this. A modern large model has two different parameter counts, and they mean completely different things.

TOTAL PARAMETERS

How much model you have to **store**. Drives memory and hardware cost.

ACTIVATED PARAMETERS

How much model **runs** on any given token. Drives compute and speed.

In a **dense** model these are the same number — Mistral Medium 3.5, at 128B dense, activates essentially all of its parameters on every token. In a **Mixture of Experts (MoE)** model they come apart, sometimes by a factor of twenty or more. Most prominent open-weight releases of the past year are MoE; not every frontier model is, and many closed models do not disclose their architecture.

## What MoE actually is

Picture a hospital. The old architecture is one enormously well-read generalist who personally sees every patient. Every question goes through the whole of that one brain. MoE is a hospital with 896 specialists and a triage desk: a question arrives, the triage desk routes it to the 16 specialists most relevant, and only those 16 do any work. The other 880 sit idle for that question — but you still have to employ all 896.

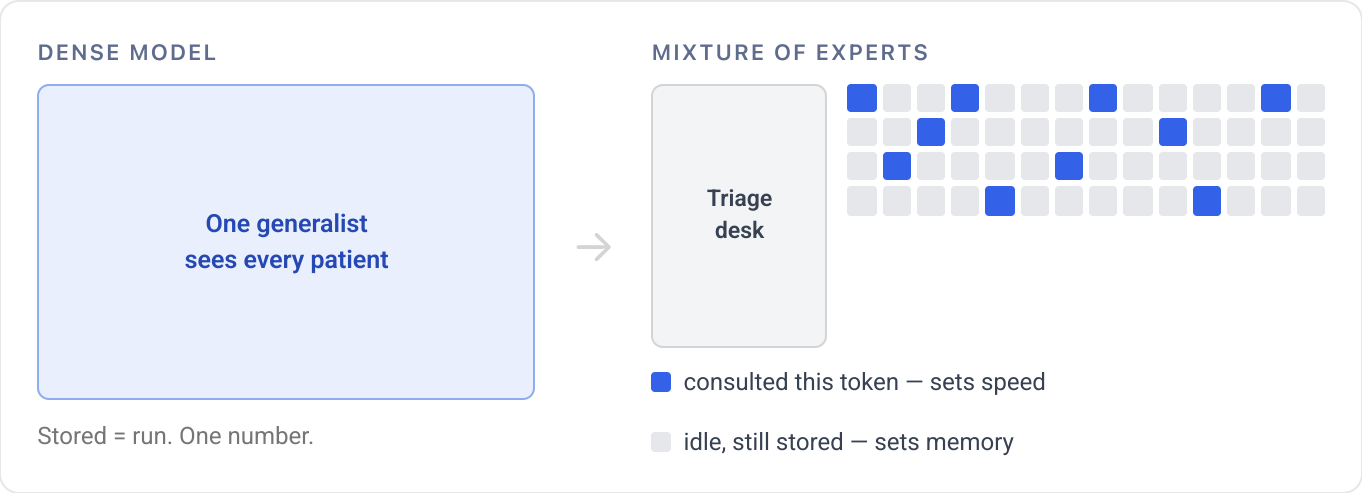



**Figure 3 —** You pay salary for the whole hospital and get the response time of a 16-person team. The analogy breaks in one place: routing is learned per token, so "specialty" is not human-interpretable.

| MODEL | TOTAL ("EMPLOYED") | ACTIVATED ("PER TOKEN") | EXPERTS | SOURCE |
|---|---|---|---|---|
| **Kimi K3** (Moonshot) | 2.78 trillion | 104.2 billion | 16 of 896 routed | arXiv 2607.24653 |
| **DeepSeek-V4-Pro** | 1.6 trillion | 49 billion | — | arXiv 2606.19348 |
| **GLM-5** (Z.ai) | 744 billion | 40 billion | 256 total | arXiv 2602.15763 |
| **DeepSeek-V4-Flash** | 284 billion | 13 billion | — | arXiv 2606.19348 |

> **You must have enough memory for the total, but you only get the speed of the activated.**

This is why "open weights" and "we can run it" have quietly become different statements. Kimi K3 is genuinely open — the weights are downloadable — and it is also, for almost every hospital system and mid-size pharma, unrunnable. Storing 2.78 trillion parameters is a multi-rack GPU problem, not a

server-closet problem, no matter how few of them fire per token.

> **WHAT TO ASK A VENDOR**
>
> Not “how big is it” but “what are the total and activated parameter counts, and what does inference cost in GPU memory at our context length.” A vendor who quotes only one number is either confused or hoping you are.

## Part 2 · The four stages of training

Every model you will evaluate went through some version of these four stages. Understanding what each stage does tells you what you can and cannot change about a model you download.

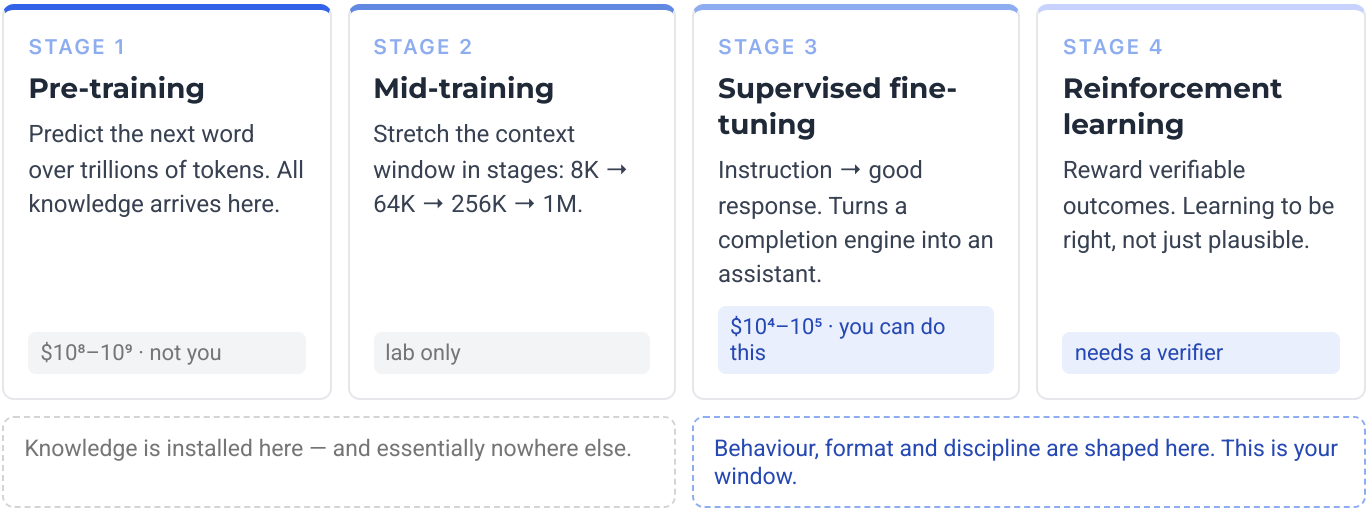



**Figure 4 —** The training pipeline, and which parts of it you can realistically touch.

### Stage 1 — Pre-training: learning language and world knowledge

The model reads an enormous amount of text and repeatedly predicts the next word. That is all it does. GLM-5 did this over 28.5 trillion tokens — an amount of text beyond any useful human-scale analogy. (At roughly 0.75 words per token and 75,000 words per book, that is on the order of 285 million books; one a day would take some 780,000 years.) Out of pure next-word prediction the model acquires grammar, facts, reasoning patterns, translation, and code. Nobody fully understands why this works as well as it does; it is an empirical result, not a designed one.

**What matters to you:** essentially all of a model’s knowledge arrives here. If a fact about oncology dosing wasn’t in pre-training, later stages will not reliably install it. Gekhman et al. (arXiv 2405.05904, EMNLP 2024) showed the mechanism directly: fine-tuning examples containing knowledge the model does not already hold are learned slowly, and once learned they increase the hallucination rate roughly linearly. Fine-tuning transfers *form* reliably and *facts* badly.

**Cost:** hundreds of millions of dollars. You will never do this. Neither will your vendor.

### Stage 2 — Mid-training: stretching the context window

The context window is how much the model can hold in mind at once. Both Kimi K3 and DeepSeek-V4 advertise 1 million tokens — roughly 750,000 words, which is dozens to a few hundred *full* clinical papers depending on length, tables and references — or a great many abstracts. Maximum accepted context is not effective context: accepting a million tokens does not demonstrate that a model integrates evidence across all of it. Models are not trained at that length from the start; it is ruinously expensive. They are stretched in stages. K3 went 8K → 64K → then a four-stage cooldown from 256K to 1M. GLM-5 extended context during mid-training through 32K, 128K and 200K stages, with progressively fewer tokens at each longer stage.

There is a detail here a careful reader should catch. Long-context benchmarks are easy to game: many "needle in a haystack" tests can be solved by attending locally, without genuinely using the window. Moonshot's stated fix is to build synthetic training tasks that cannot be solved without attending across the whole million-token window. That is the honest engineering answer, and it is a good question to put to any vendor claiming long context: *how do you know the model uses the window rather than the edges of it?*

### Stage 3 — Supervised fine-tuning (SFT): learning to be useful

The base model out of pre-training is a text-completion engine, not an assistant. Ask it a question and it may produce more questions, because that is what follows a question in its training data. SFT trains on examples of instruction → good response. This is the cheap stage — thousands to millions of examples rather than trillions of tokens — and it is the stage most organizations can actually perform.

A finding worth internalizing: most of what SFT does is **elicitation, not teaching**. Hewitt et al., *Instruction Following without Instruction Tuning* (arXiv 2409.14254), showed you can produce instruction-following behavior by tuning on responses with the instructions removed entirely — and even with a hand-written rule that adjusts three probabilities. The capability was already sitting in the base model; post-training is a thin wrapper that surfaces it. This cuts both ways. Good news: adapting an open model to your clinical formats and conventions is genuinely tractable. Bad news: fine-tuning is far less reliable for *installing* unfamiliar facts than for eliciting capability the base model already has. Gekhman et al. found unfamiliar factual examples were learned slowly and raised hallucination propensity as they were learned. For mutable or safety-critical facts, retrieve and verify rather than assuming tuning baked them in.

### Stage 4 — Reinforcement learning: learning to be right

SFT teaches imitation of good answers. RL teaches the model to find good answers by rewarding outcomes. The landmark here is DeepSeek-R1 (Nature 645, doi:10.1038/s41586-025-09422-z), the first major LLM to pass peer review in a top journal — it took the cover in September 2025. The startling result comes specifically from **DeepSeek-R1-Zero**, the pure-RL variant. Starting from DeepSeek-V3

Base with no SFT cold start, the authors applied GRPO using rule-based *accuracy and format* rewards, and reported AIME-2024 average pass@1 rising from 15.6% to 77.9% during RL. The model developed its own strategies along the way, including self-checking and backtracking. The final shipped DeepSeek-R1 used a broader multi-stage recipe with cold-start data — so "no human demonstrations at all" describes R1-Zero, not R1.

Why this matters for clinical work: RL works where answers are verifiable. Math has a right answer. Code either runs or doesn't. This is why every current model is strong at math and code and comparatively weaker at open-ended judgment — those are the domains where the reward signal is clean. GLM-5 makes the sequence explicit: reasoning RL across math, science, code and tool-integrated reasoning, then agentic RL on coding and search, then general RL. K3 runs RL across general tasks, general agents and coding agents, at three separate reasoning-effort levels so the model learns to spend more compute on harder problems.

> THE CLINICAL READ
>
> Ask whether a model was RL-trained on anything resembling your task. "Strong at agentic coding" does not transfer to "reliable at adverse-event extraction." The reason is structural — one has a verifier and the other doesn't.

## Part 3 · Distillation — how a small model gets good

This is the stage most relevant to anyone who wants something they can actually host, and for most organizations it is the fastest and most cost-effective route to a useful small model.

Distillation transfers useful behavior from a stronger **teacher** to a smaller **student**. The student does not copy the teacher's weights. It is trained to reproduce the teacher's behavior on a selected set of tasks. In the simplest and most practical form — **behavior cloning** — you give the teacher representative tasks, collect high-quality answers, and train the student on those input–output pairs. At the implementation level this looks like supervised fine-tuning: the student learns from demonstrations of the desired behavior exactly as it would learn from human-written ones.

A point that gets lost in the vendor framing: **the teacher does not have to be a model.** It can be a stronger model, a group of human experts, or both. Throughout this section "teacher" means any reliable source that can demonstrate the behavior you want. Human experts may write ideal answers, review and correct model-generated ones, or define the rules and edge cases the student should follow. The most effective arrangement is usually hybrid — a stronger model supplies scale, human experts define and control the standard.

## The practical recipe

1. **Collect tasks**
   Assemble representative examples from the workflow you want to automate.
2. **Create demonstrations**
   Have a stronger model or human experts produce high-quality answers.
3. **Review the outputs**
   Correct errors, resolve disagreements, remove unsafe examples.
4. **Train the student**
   Fine-tune a smaller model to reproduce the demonstrated behavior.
5. **Evaluate**
   Test on a held-out, independently adjudicated evaluation set.

Steps 2 and 3 are the expensive ones — not step 4.

**Figure 5 —** Behavior cloning end to end. The cost sits in producing examples that accurately represent the task, not in the training run.

Human experts can write those examples directly, but expert time is the scarcest input you have. A stronger model generates candidate answers quickly; experts then spend their attention on reviewing difficult cases, resolving disagreements and defining what "correct" means. That shifts the human contribution from writing every answer to **designing the task, reviewing the outputs and handling the cases where judgment matters** — which is where clinician time is actually worth spending.

## Why the return on investment is high

Distillation concentrates broad capability into a narrow deployment target. A large teacher must perform well across many domains. Your student may only need to perform a handful of clinical workflows — it does not need to reproduce the teacher everywhere, only where your organization will actually use it. Three economic consequences follow.

DEVELOPMENT
### Fast
You start from a capable student and a set of strong demonstrations — not from pre-training.

INFERENCE
### Cheap
A trained student may run on a single workstation GPU where the teacher needs a cluster.

DEPLOYMENT
### Controllable
Weights can be pinned, hosted inside your data boundary and validated against a fixed configuration.

The teacher — model, experts, or both — supplies the desired behavior during development. The student makes that behavior cheap to deploy, repeatedly. That division is the whole economic case.

### What actually transfers

When the teacher is a model rather than a person, three distinct mechanisms hide under the one word — and what you can access depends entirely on whether you hold the teacher's weights.

| TECHNIQUE | WHAT TRANSFERS | AVAILABLE FROM A FRONTIER API? |
|---|---|---|
| **Logit distillation** | The teacher's full probability distribution over the next word — including how it ranked the wrong answers | No |
| **Behavior cloning** | The teacher's full output including its reasoning trace | No — frontier reasoning models return summaries |
| **Behavior parroting** | Final answers only | Yes |

Hinton, Vinyals & Dean's original insight — *Distilling the Knowledge in a Neural Network* (arXiv 1503.02531, 2015) — was that **the wrong answers carry the information**. A teacher that says "this adverse event is 70% cardiac, 25% respiratory, 5% hepatic" teaches vastly more than a label saying "cardiac" — it teaches what the confusable neighbors are. That richness is why one soft-labeled example is worth many hard-labeled ones, and why the top row of that table is so much stronger than the bottom.

### Start simple, then improve: on-policy distillation

More sophisticated forms exist, and they are not where most organizations should start. If the teacher is another model, the student can learn from more than final answers. The highest-return first step is still conventional behavior cloning — define the real task, create high-quality demonstrations, fine-tune a hostable student, evaluate against an independent clinical benchmark, then improve the dataset where the student actually fails. **Add more complicated training machinery only after that baseline has been measured.** What follows is what you graduate to, not what you begin with.

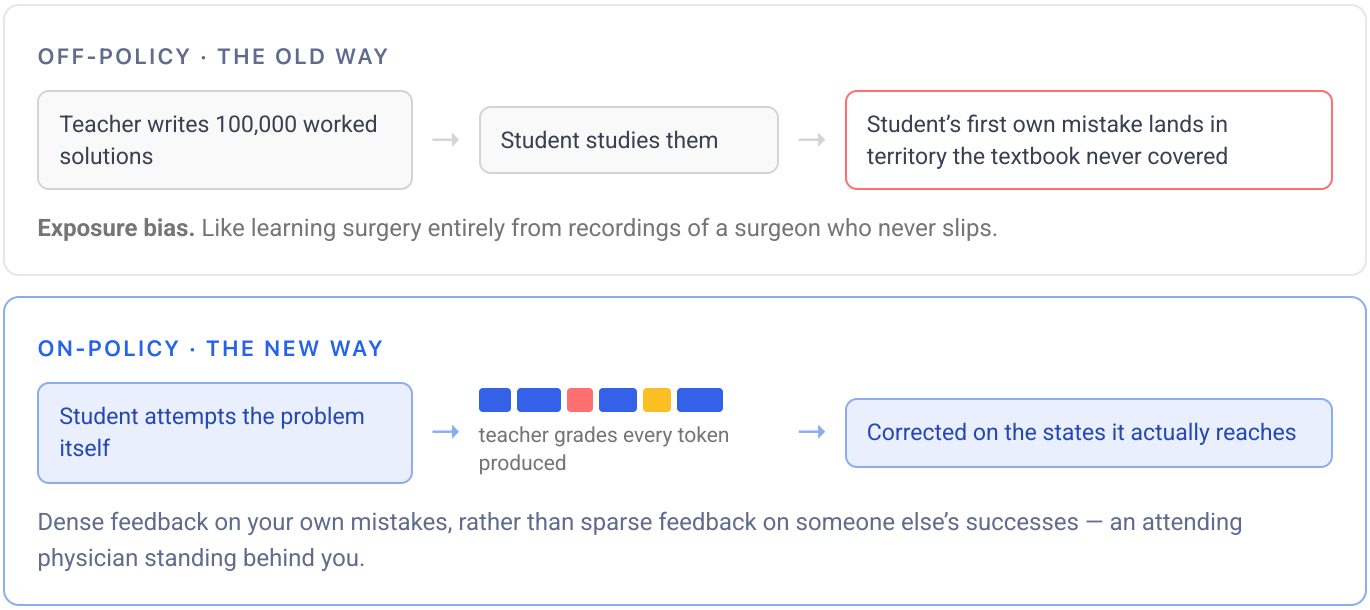



**Figure 6 —** Why on-policy distillation works: the correction signal lands where the student actually is.

This is not a research curiosity. Per the field survey (arXiv 2604.00626) it is now standard post-training at Alibaba (Qwen3), DeepSeek (V4), Z.ai (GLM-5), Xiaomi (MiMo) and NVIDIA (Nemotron). Thinking Machines Lab reproduced the recipe on Qwen3-8B with Qwen3-32B as teacher at a fraction of RL compute. For the theoretical account, *On the Geometry of On-Policy Distillation* (arXiv 2606.07082) shows OPD occupies a distinct parameter-space regime rather than interpolating between SFT and RL.

K3 goes a step further with **multi-teacher on-policy distillation**, consolidating nine separate expert models into the final one — train nine specialists, then merge them into a single generalist. GLM-5 uses on-policy distillation for a different purpose: preventing capability regression between training stages, so improving agentic skill doesn't quietly degrade reasoning.

The Bridgewater case in Part 0 used exactly this, with one refinement worth stealing: every 20 steps the current checkpoint was promoted to teacher, but only if validation accuracy had reached a new high — so training never distilled toward a weaker model. Note what that implies about the word "teacher." By the end, the student's teacher was an earlier version of itself, and what it was really learning from was the data.

### When on-policy distillation fails

The most useful recent paper here is *Rethinking On-Policy Distillation* from Tsinghua's NLP group (arXiv 2604.13016), and it says something practical: OPD is not free. It requires two conditions.

1. **Compatible thinking patterns.** Teacher and student must reason in broadly similar shapes. Too large a stylistic gap and the dense token-level feedback is noise.

2. **The teacher must have something genuinely new to offer.** A higher benchmark score is not sufficient. If the teacher merely does what the student already does, slightly better, you gain nothing.

Their diagnostic is elegant: in successful runs, teacher and student converge on a small shared set of tokens carrying 97–99% of the probability mass at the states the student visits. When that overlap is low at the start, training fails. Their fix — directly actionable — is an **off-policy cold start**: a short conventional SFT warmup on teacher outputs to raise the initial overlap, then switch to on-policy. If you build a clinical specialist from a general teacher, you are in exactly the mismatched-pattern regime this describes, and the cold start is the prescribed remedy.

THE LIMIT NOBODY SHOULD SKIP

**Distillation raises floors. It does not raise ceilings.**

Gudibande et al., *The False Promise of Imitating Proprietary LLMs* (arXiv 2305.15717), ran the direct experiment: students from 1.5B to 13B trained on up to 150M tokens of ChatGPT output. Human raters judged them roughly competitive with the teacher. Targeted benchmarks showed they had closed almost none of the real gap. Imitation transfers style more than capability — and human preference evaluation hides the difference. If your evaluation of a distilled clinical model is "our reviewers liked the outputs," you have reproduced this experiment and will get its result.

THE CLINICAL READ

Distillation is usually the shortest path from "we can demonstrate this task well" to "we have a smaller model that performs it affordably and consistently." But the value does not come from collecting a large volume of attractive answers. It comes from choosing representative tasks, controlling the quality of the demonstrations, and evaluating the student independently.

A stronger model gives you scale. Human experts give you standards and judgment. The benchmark tells you whether the student learned anything real.

## Part 4 · Three efficiency terms, and what each buys

**Sparse attention.** Normally every word attends to every other word, so doubling the input quadruples the work — this is why long context is expensive. GLM-5 adopts DSA (DeepSeek Sparse Attention, originally from DeepSeek-V3.2, arXiv 2512.02556), which looks at the content to pick which tokens matter rather than using a fixed pattern, cutting attention compute roughly 1.5–2× on long sequences. DeepSeek-V4 combines Compressed Sparse Attention and Heavily Compressed Attention to reach a more dramatic result: at 1M-token context, 27% of the per-token inference FLOPs and 10% of the KV

cache of V3.2. K3 takes a different route — Kimi Delta Attention, a linear-attention variant, interleaved with Gated MLA full-attention layers at a 3:1 ratio (three cheap layers, one expensive one), reporting roughly 2.5× better overall scaling efficiency than K2.

That KV cache number is the one to care about operationally. The **KV cache** is the model's working memory for the current conversation, it lives in GPU RAM, and at long context it often exceeds the model weights themselves. A 10× reduction is the difference between a long-context workload being feasible on your hardware and not.

**Quantization.** Model weights are stored at reduced numerical precision to fit in less memory. K3 ships expert parameters at MXFP4 weights and MXFP8 activations while keeping non-expert components at higher precision — because the experts are where the parameter mass is, and the rest is where precision-sensitivity lives. Quantization is how a model that nominally needs N GPUs fits on fewer, with some quality cost. Always ask what precision a quoted hardware requirement assumes.

**Speculative decoding / multi-token prediction.** GLM-5 uses MTP with parameters shared across 3 layers. A small fast model guesses several tokens ahead and the big model verifies them in one pass. Pure speed optimization; no quality change. Worth knowing only because it appears in throughput claims.

## Part 5 · What it takes to run one

Everything so far has been about how models are made. This part is about what happens when you try to run one, because that is where most "let's go local" plans meet arithmetic. The models in the news are the trillion-parameter ones. The models you would actually deploy are mostly not.

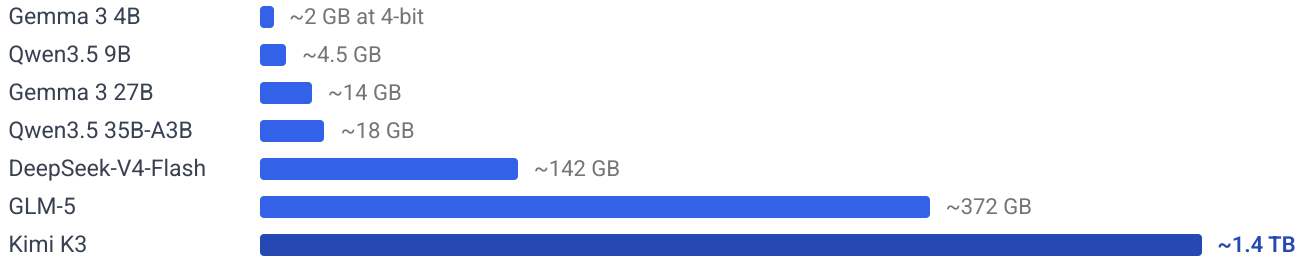
| TIER | REPRESENTATIVE MODELS | TOTAL | ACTIVATED |
|---|---|---|---|
| Edge | Gemma 3 1B / 4B · Qwen3.5 0.8B / 2B / 4B · Ministral 3B | 1–4B | same (dense) |
| Workstation dense | Qwen3.5 9B · Gemma 3 12B / 27B | 7–32B | same (dense) |
| Workstation MoE | Qwen3.5 35B-A3B | ~35B | ~3B |
| Large MoE, single node or more | Mistral Small 4 · DeepSeek-V4-Flash | 119–284B | 6.5–13B |
| Large dense | Mistral Medium 3.5 (**dense**) | 128B | ~128B |
| Multi-GPU MoE | GLM-5 / GLM-5.2 | 744B | 40B |
| Cluster only | DeepSeek-V4-Pro · Kimi K3 · Qwen3.8-2.4T-A95B | 1.6–2.8T | 49–104B |

Licences belong to specific checkpoints, not to hardware tiers, so they are deliberately not in this table — read the `LICENSE` file of the exact artifact. Note the two Mistral rows: **119B activating 6.5B** versus **128B activating all 128B** — near-identical totals, roughly twentyfold difference in per-token work.

Treat this table as a snapshot, not a reference. This tier of the field re-releases every few weeks and much of what is written about it online is marketing. Verify sizes and licences on the model card before acting on any row. Note the third row especially: Qwen3.5 35B-A3B holds 35 billion parameters and activates 3 billion — workstation-class memory at the speed of something five times smaller. MoE is not only a frontier-lab technique.

### RULE OF THUMB 1 · MEMORY FOLLOWS TOTAL PARAMETERS

Raw weight payload ≈ total parameters × bits per parameter ÷ 8 — a lower bound, not a VRAM budget. 4-bit ≈ 0.5 GB per billion · 8-bit ≈ 1 GB · BF16 ≈ 2 GB. Real checkpoints also carry quantization metadata, higher-precision tensors and runtime workspaces: Google's Gemma 3 int4 needs ~2.6 GB for the 4B and ~14.1 GB for the 27B just to load. Then add KV cache.




Bars are indicative, not linear — Kimi K3 runs off the chart and the smallest bars are floored so they stay visible. Then add the KV cache, which is the part people forget: it scales with context length and concurrency, and at long context it routinely exceeds the weights.

### RULE OF THUMB 2 · SPEED FOLLOWS ACTIVATED PARAMETERS

```
tokens/second ≈ memory bandwidth ÷ (activated params × bytes per param)
```

Prefill (reading your prompt) is generally compute-bound; decode (emitting tokens) is generally memory-bandwidth-bound at low batch size, because each token requires reading the active weights out of memory. Which is why MoE decodes quickly — only the activated slice gets read. Treat this as a roofline ceiling, not a throughput predictor: it ignores shared dense layers, attention and KV traffic, routing, interconnect, batching and kernel efficiency.

**Figure 7 —** The two independent dials: one sets whether the model fits, the other sets how fast it answers.

| HARDWARE | VRAM | BANDWIDTH | 9B DENSE | 35B-A3B (3B ACTIVE) |
|---|---|---|---|---|
| RTX 4090 | 24 GB | ~1.0 TB/s | ~200 tok/s theoretical, 60–100 real | fits; ~250+ theoretical |
| L40S | 48 GB | ~0.86 TB/s | ~50–80 real | comfortable |
| A100 80GB | 80 GB | ~2.0 TB/s | ~120–180 real | fast |
| H100 80GB | 80 GB | ~3.35 TB/s | ~200–300 real | very fast |
| H200 141GB | 141 GB | ~4.8 TB/s | — | — |
| 8×H100 node | 640 GB | — | serves a 284B/13B MoE | — |

Real throughput lands well below the theoretical ceiling — attention overhead, sampling, framework inefficiency and imperfect memory access all take their cut. Use theoretical as an upper bound and expect roughly a third to a half of it single-stream. Batching many concurrent requests raises total

throughput substantially while leaving per-request speed roughly flat, which is exactly why serving providers can be cheaper per token than you.

- A 9B model on one workstation GPU generates faster than most people read. Capability, not speed, is the constraint at that size.
- Kimi K3's raw 4-bit weight payload is ~1.4 TB, so **at least eighteen 80 GB H100s just to hold it** (18 × 80 GB = 1.44 TB) — leaving essentially nothing for mixed-precision tensors, metadata, workspaces or KV cache. A production deployment needs meaningfully more. That is the arithmetic behind Part 1's claim that open weight and runnable have become different properties.

### The remark that saves most teams a year: don't self-host first

If your reason for wanting open weights is control over *which model and which version* — not control over the physical machine — you can have almost all of it without buying a GPU.

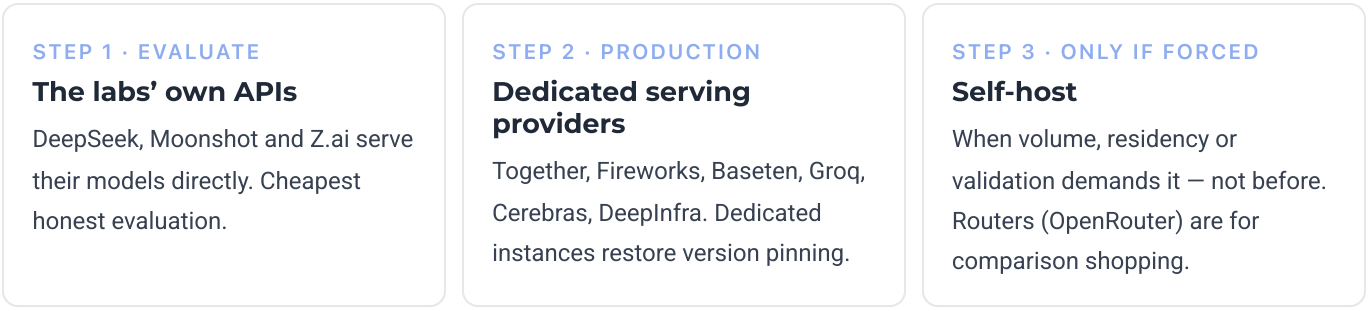



**Figure 8 —** The sequence that works. Buying hardware first is how teams end up with an expensive cluster at 15% utilization and no evaluation set.

Three caveats specific to regulated work, and the first is the one that catches people.

- **A router's *default* is not one endpoint.** Left alone, a router may load-balance across providers, endpoint variants and fallback paths, so requests can be served at different quantizations on different days. This is configurable rather than inherent: pin the provider, filter quantization, disable fallbacks, require zero-retention endpoints, and log the routing metadata with every request. Two providers serving "the same" model at 4-bit and 8-bit will not give you the same outputs, and neither is wrong. For exploratory work this is fine; for anything you must reproduce it is disqualifying unless you pin the provider and the quantization.
- **The data boundary comes back.** A hosted provider re-creates every question you were trying to avoid — who sees the payload, who sees the query pattern, what the retention terms are. Open weights served by someone else is a licensing and model-choice win, not a data residency win.
- **Ask what quantization is being served.** It is frequently unstated and it materially changes quality. A model that underperforms on your evaluation may be a 4-bit copy of one that would have passed.

## Part 6 · Licences — the part that isn't in the benchmark table

"Open weights" is a range, not a category. Two models can both be "open" and give you very different rights.

| MODEL | WEIGHTS LICENCE | COMMERCIAL USE | FINE-TUNING | CONDITIONS |
|---|---|---|---|---|
| Kimi K3 | Custom "Kimi K3 License" (modified MIT) | Yes | Yes | Separate agreement required if Model-as-a-Service revenue exceeds $20M/12 months; must display "Kimi K3" in UI above 100M MAU or $20M monthly revenue |
| GLM-5 / GLM-5.2 / GLM-5.3 | MIT | Yes | Yes | Standard MIT terms; no regional restrictions, no revenue trigger |
| DeepSeek-V4-Pro / V4-Flash | MIT (code and weights) | Yes | Yes | The most permissive of the frontier-scale options — no MAU cap, no notification, no attribution. Older DeepSeek releases (V3 base, Coder-V2, VL2) split a permissive code licence from a more restrictive weights licence; check per release |
| Qwen3.6-35B-A3B and other open Qwen releases | Apache 2.0 | Yes | Yes | Includes an express patent grant — the one thing MIT does not give you. But the family is mixed: Alibaba's strongest current models (Qwen3.7-Max, Qwen3.6-Plus, Qwen3.5-Omni) are API-only with no published weights. Do not assume "Qwen" means open |
| Gemma 4 (Google) | Apache 2.0 | Yes | Yes | Google moved Gemma 4 to plain Apache 2.0, dropping the separate Gemma-terms agreement that Gemma 3 required. If you are evaluating a Gemma 3 checkpoint, you are still under the older custom terms |
| Nemotron 3 (NVIDIA) | NVIDIA Nemotron Open Model License | Yes | Yes | Permissive in substance — commercial use, derivatives, and NVIDIA claims no ownership of outputs — but it is a bespoke licence, not OSI-approved, and lawyers have flagged it as such. Attribution notice required when redistributing. Some Nemotron checkpoints are built on Gemma and inherit Gemma Terms of Use on top |

| MODEL | WEIGHTS LICENCE | COMMERCIAL USE | FINE-TUNING | CONDITIONS |
|---|---|---|---|---|
| Muse Glimmer (Meta) | Apache 2.0 | Yes | Yes | Meta's current open release (10 August 2026): 30B dense multimodal, 128K context, ungated weights, under 20 GB at 4-bit — a workstation model, not a frontier one. Note the change of posture: the older Llama 4 line sits under the restrictive Llama Community License (700M MAU cap, "Built with Llama" attribution, multimodal capabilities not licensed to EU-headquartered companies), so do not carry Llama-era licence assumptions onto Muse, or vice versa. Meta has said the closed Muse Spark 1.2 weights will be opened; as of mid-August 2026 they had not shipped |

All of these are usable, and they are not equally usable. Three patterns are worth naming. **Plain MIT or Apache 2.0** — DeepSeek V4, GLM-5, the open Qwen releases, Gemma 4, Muse Glimmer — is the clean case: no thresholds, no attribution, nothing that changes as you grow. Apache 2.0 adds an express patent grant that MIT lacks, which is the one substantive difference between them for a commercial deployment. **Bespoke "open" licences** — Kimi K3, Nemotron — grant broad rights but on their own drafted terms: K3's $20M Model-as-a-Service revenue threshold converts an open licence into a negotiation, and adds a UI branding requirement at scale. Neither would bind a clinical research group any time soon; both would matter a great deal to a platform business planning to serve the model at volume. **Community licences** — Meta's older Llama line, Gemma 3 — are the ones to read most carefully, because the restrictions are field-of-use and jurisdictional rather than merely commercial, and a clause like Llama 4's EU multimodal carve-out does not show up anywhere in a benchmark comparison. Both vendors have since moved their current releases to Apache 2.0, which is the direction of travel but not retroactive.

Two practical notes. Licence terms follow the *checkpoint*, not the brand: Gemma 4 and Gemma 3 differ, DeepSeek V4 and V3 differ, and "Qwen" spans Apache 2.0 and fully proprietary. And a fine-tune inherits its base model's licence, which means the terms above govern the clinical specialist you build, not just the weights you download.

> Check the licence **before** the technical evaluation, and read the `LICENSE` file in the weights repository — the only authoritative source. Revenue triggers, attribution at scale, field-of-use limits, and who owns a fine-tune you produce live in that file, not in the announcement.

## Part 7 · Reading a release: a worked example

Kimi K3 announces: 2.8T parameters, 104B activated, 1M context, tops the open models, weights on HuggingFace. Here is the same announcement with the three questions applied to it.

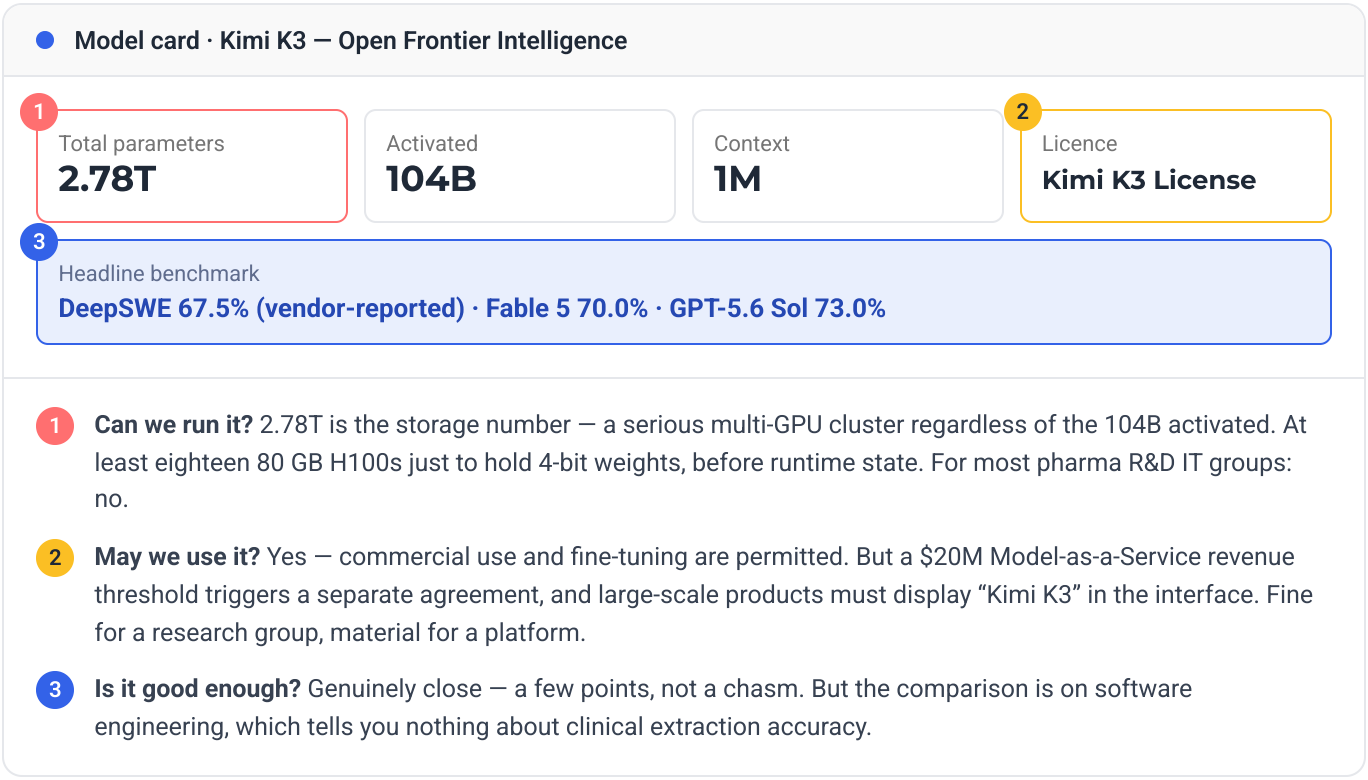



**Figure 9 —** The same release, annotated. Every number on a model card belongs to one of the three questions; the announcement will not tell you which.

So the honest summary of the most celebrated open release of 2026, for a commercial clinical team: **impressive, licensable, effectively unrunnable at your scale, and benchmarked on something other than your task.** The binding constraint turned out to be hosting, not licensing. That is an argument for reading past the headline number — GLM-5 at 744B under MIT, or DeepSeek-V4-Flash at 284B total and 13B activated, are far more relevant to you despite lower headline scores, because you can actually serve them.

The general pattern: the frontier open models have converged on architectures optimized for inference at scale in the lab's own datacenter, not for portability into yours. Total parameters keep climbing because MoE makes it cheap to serve — for someone running thousands of concurrent requests across a big cluster. That economic logic is not your logic.

## Part 8 · Domain tuning, and what actually holds value

The obvious move for a clinical organization is to take an open model and tune it on medicine. It is a good move. It is also more perishable than most people assume, and being clear-eyed about which parts perish is what makes the investment sound.

Domain tuning buys three things reliably: **vocabulary and conventions** (the model stops paraphrasing SOC terms and starts using them), **output discipline** (it produces your extraction schema every time rather than most times), and **task accuracy** on work the base model could nearly already do. One thing it does not buy: new medical knowledge. In clinical work the facts in question are dosing, endpoints, eligibility and safety signals — precisely the wrong list to be shaky on. **Tune for form; retrieve for facts.**

| ACTIVITY | ORDER OF MAGNITUDE (ILLUSTRATIVE ESTIMATE) | WHO CAN DO IT |
|---|---|---|
| Pre-training a frontier model | $10^8$–$10^9$ | Frontier labs only |
| Continued pre-training on a domain corpus | $10^5$–$10^6$ | Well-funded organizations |
| SFT / on-policy distillation on an open base | $10^4$–$10^5$ | A competent team |
| Building the evaluation set that tells you if it worked | $10^4$–$10^5$ and months of clinician time | Very few |

Note what that last row does to the analysis. The training run is no longer the expensive part. The adjudicated evaluation set is — and unlike the training run, you cannot buy it, rent GPUs for it, or finish it in a weekend.

### The treadmill, and why it's less bad here than elsewhere

Frontier and open base models refresh roughly every six to twelve months, and each new base arrives having absorbed some of what your fine-tune was adding. A domain model is a depreciating asset. Continuous re-tuning is the answer, and it is economically rational under three conditions: the pipeline is automated; the evaluation set is stable and reusable, so "is the new base already better than our tuned old one?" is a cheap question; and the measured increment is still material. Miss the second and the treadmill becomes ruinous, because you cannot tell whether to step onto it.

Now the twist that makes regulated work different. In consumer products the treadmill is relentless because a stale model is a competitive liability. In regulated work the incentive to update is much weaker. Current thinking — ICH E6(R3) and FDA's risk-based framework — does not demand full revalidation whenever anything changes; it demands **change control**. A change to weights,

quantization, tokenizer, prompt template, retrieval corpus, tool configuration, serving engine, dependency or hardware enters a documented impact assessment, and the scope of regression testing is proportionate to risk. That is less absolute than “changing the model means revalidating,” but points the same way: every upgrade carries an assessment cost a consumer product never pays. My inference — a strategic judgment, not a regulatory requirement — is that pinned weights are therefore an asset here.


**Figure 10 —** A competitor with the same open base and a good engineer reproduces your fine-tune in weeks. Nothing about the weights is defensible; what surrounds them is.

Four caveats worth stating plainly. **Catastrophic forgetting:** tuning hard on one domain degrades general ability — GLM-5’s use of on-policy distillation specifically to prevent cross-stage regression is the industrial acknowledgment of this. **Synthetic-data collapse:** training recursively on generated data erases the tails of the distribution first, then degrades the mode (Shumailov et al., Nature 631, 755–759, 2024). **Evaluation that hides failure:** reviewer preference is the instrument that will fail you. **Teacher terms of service:** several major providers restrict using outputs to train competing models — distilling from a frontier teacher into a product you then license is exactly what that clause targets.

**The honest framing for anyone deciding where to spend: the weights depreciate and the evaluation appreciates.** A team that spends a year building a beautiful domain model and no evaluation set owns a wasting asset. A team that spends the year building adjudicated gold standards and a verification harness can re-tune onto whatever base is best six months from now, cheaply, and know that it improved.

## Case study: benchmark-first development in our group

> **Disclosure.** I am an author or contributor on the resources below, produced by my research group and company. Read this as a worked example of the argument above, not as independent corroboration of it.

This is not an abstract recommendation. It describes where our own group has put its effort for several years, and the artifacts are public — which is the point, since a benchmark nobody can run against you is a marketing claim rather than a measurement.

| BENCHMARK | WHAT IT MEASURES | SCALE | VENUE |
|---|---|---|---|
| PyHealth 2.0 | Reproducible clinical deep learning — standardized pipelines so results can be re-run rather than re-asserted | Toolkit plus benchmark suite | ICML 2026 · arXiv 2601.16414 |
| PyTrial | 34 ML algorithms across 6 clinical trial tasks — outcome prediction, site selection, patient–trial matching, similarity search, synthetic data generation | 6 task families | arXiv 2306.04018 |
| CTO | Trial success/failure labels fused from registries, publications, phase progression, news sentiment and sponsor stock movement, plus a manually annotated 2020–2024 set | ~125,000 trials; F1 94 on Phase 3, 91 across all phases | Nature Health (2026) · arXiv 2406.10292 |
| TrialBench | Multi-modal AI-ready datasets spanning trial design and operations | 23 curated datasets | Scientific Data · arXiv 2407.00631 |
| MLOmics | Cancer multi-omics across supervised, unsupervised and imputation tasks | 8,314 samples, 32 cancer types, 4 omics types | Scientific Data 12 (2025) · arXiv 2409.02143 |
| BioDSA-1K | Data science agents on biomedical hypothesis validation — decision accuracy, evidence–conclusion alignment, reasoning correctness, code executability | 1,029 hypothesis tasks, 1,177 analysis plans, 300+ studies | arXiv 2505.16100 |
| TrialPanorama | Systematic review (search, screening, evidence summarization) and trial design (arm design, eligibility, endpoints, sample size, completion) | 1.6M trial records from 15 registries; 152K samples across 8 tasks | npj Digital Medicine (2026) · arXiv 2505.16097 |

Two features of that list are the transferable lessons. **It includes tasks with no clean answer.** BioDSA-1K deliberately contains non-verifiable hypotheses — cases where the available data are insufficient to support or refute the claim. Most benchmarks quietly exclude these because they are hard to score; excluding them trains and measures for confident answering, which is the single worst failure mode in clinical work. If your evaluation set has no “the data cannot answer this” category, it is rewarding the behavior you least want. **It measures the process, not only the output.** BioDSA-1K scores whether the

reasoning was correct and whether the generated code actually executes, separately from whether the final answer was right. A model that reaches the right conclusion through invalid analysis has failed, and only a benchmark built this way can see it.

Note the dependency, because it is the whole argument of this section: the 8B result in Part 0 was only possible because the benchmark existed first. The evaluation was not a way of checking the model — it was the thing that made the model buildable, and the same is true of the expert labels in the finance case. Public benchmark results for our own systems are at keiji.ai/research/benchmarks, and we contribute to open benchmark development through the R Consortium — a benchmark you publish is one a customer can run against you, which is a much stronger claim than a number in a slide deck.

## Part 9 · Seven conclusions I would defend

01 **"Open source caught up" is two true clauses joined by a false therefore.** Open models have closed much of the capability gap. Open models you can realistically host — the 8B to 70B range — have not. Both statements are true and they are about different models.

02 **Licence is a gating criterion, not a footnote.** Check it before the technical evaluation, and read the LICENSE file in the weights repository rather than the licence shown on the paper. Open-weight terms are increasingly bespoke.

03 **Capability is task-shaped.** Near-parity on extraction, structuring, classification and retrieval-grounded work; a real gap on long-horizon agentic planning and multi-step tool use. Fortunately the layer that must run locally — execution over patient records — is largely the layer where small models are closest to parity.

04 **Distillation is the right tool, with a known failure mode.** On-policy distillation is how you build a hostable clinical specialist. It requires a cold start when teacher and student reason differently, and it will not install knowledge the base model lacks. Evaluate against targeted benchmarks, never against reviewer preference.

05 **Frozen weights are a scientific requirement before a security one.** A trial runs five to ten years; frontier models are deprecated in twelve to eighteen months. You can pin a version on a hyperscaler, but you don't set the retirement date. And frozen weights alone do not give bit-reproducible output — MoE routing and kernel non-determinism see to that. Local doesn't fix reproducibility for free; it makes it achievable, via fixed seed, fixed batch, pinned kernels, known hardware. None of that is engineerable through a shared endpoint.

06 **Distillation from a frontier teacher has a contract problem.** Most frontier providers prohibit using outputs to train competing models. Distilling from a frontier teacher into a model you then license to pharma is precisely what those clauses target. This is a legal review item before it is a training plan.

07 **Specialized beats generic on narrow work, and it beats it cheaply.** Two industries, one result: a fine-tuned open-weight model beat every frontier model tested on six financial-judgment tasks at 1/13.8th the cost, and an 8B clinical model beat 70B generic models on all eight TrialPanorama tasks. Neither needed a new algorithm — both needed data nobody else had. So the strategic question is not which frontier model to license. It is which of your organization's judgments is worth labeling.

# Glossary

| TERM | PLAIN MEANING |
|---|---|
| **Token** | Roughly ¾ of a word; the unit models read and write |
| **Parameters** | The learned numbers that constitute the model |
| **Activated parameters** | The subset that runs per token — sets speed |
| **Total parameters** | All of them — sets memory and hardware cost |
| **MoE** | Mixture of Experts: many specialists, few consulted per token |
| **Context window** | How much text the model holds at once |
| **KV cache** | Working memory for the current conversation; often exceeds weight size at long context |
| **Pre-training** | Learning language and knowledge from vast text |
| **SFT** | Supervised fine-tuning: learning to follow instructions |
| **RL** | Reinforcement learning: learning from outcome rewards |
| **Distillation** | Training a small model to behave like a large one |
| **On-policy distillation** | Teacher grades the student's own attempts, token by token |
| **Exposure bias** | Student never learns to recover from its own errors |
| **Quantization** | Lower numerical precision to save memory. 4-bit ≈ 0.5 GB per billion parameters |
| **Sparse attention** | Attending selectively rather than to everything |
| **Memory bandwidth** | How fast the GPU reads its own memory. Sets generation speed — decoding is bandwidth-bound, not compute-bound |
| **Serving provider** | A company that hosts open weights for you (Together, Fireworks, Baseten, Groq, DeepInfra) |
| **Router** | An API above many providers (e.g. OpenRouter). One interface, but not one endpoint — quantization and provider can vary per request |

# References


### MODEL REPORTS

- Kimi K3: Open Frontier Intelligence — arXiv 2607.24653 (Moonshot AI). Weights under the custom Kimi K3 License.
- GLM-5: from Vibe Coding to Agentic Engineering — arXiv 2602.15763 (Z.ai). Weights under MIT.
- DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence — arXiv 2606.19348.
- DeepSeek-V3.2 — arXiv 2512.02556 (origin of DSA sparse attention).
- DeepSeek-R1 — Nature 645, doi:10.1038/s41586-025-09422-z (cover, September 2025).
- mHC: Manifold-Constrained Hyper-Connections — arXiv 2512.24880 (DeepSeek).

### DISTILLATION AND POST-TRAINING

- Hinton, Vinyals & Dean, Distilling the Knowledge in a Neural Network — arXiv 1503.02531 (2015).
- A Survey of On-Policy Distillation for LLMs — arXiv 2604.00626.
- Rethinking On-Policy Distillation: Phenomenology, Mechanism, and Recipe — arXiv 2604.13016 (THUNLP) — the cold-start prescription.
- On the Geometry of On-Policy Distillation — arXiv 2606.07082.
- Thinking Machines Lab, On-Policy Distillation — thinkingmachines.ai.
- Su, Zhu, Xiao, Alur & Kang (Bridgewater AIA Labs) with Thinking Machines Lab, Learning to Replicate Expert Judgment in Financial Tasks — June 2026.

### LIMITS OF FINE-TUNING AND IMITATION

- Hewitt et al., Instruction Following without Instruction Tuning — arXiv 2409.14254.
- Gekhman et al., Does Fine-Tuning LLMs on New Knowledge Encourage Hallucinations? — arXiv 2405.05904 (EMNLP 2024).
- Gudibande et al., The False Promise of Imitating Proprietary LLMs — arXiv 2305.15717.
- Shumailov et al., AI models collapse when trained on recursively generated data — Nature 631, 755–759 (2024), doi:10.1038/s41586-024-07566-y (Author Correction doi:10.1038/s41586-025-08905-3).

### BENCHMARKS FROM OUR GROUP

- PyHealth 2.0 — ICML 2026, arXiv 2601.16414 · github.com/sunlabuiuc/PyHealth.
- PyTrial — arXiv 2306.04018.
- CTO — Nature Health (2026), doi:10.1038/s44360-026-00081-6 · arXiv 2406.10292.
- TrialBench — Scientific Data, arXiv 2407.00631.
- MLOmics — Scientific Data 12 (2025), arXiv 2409.02143.
- BioDSA-1K — arXiv 2505.16100 · biodsa.github.io.
- TrialPanorama — npj Digital Medicine (2026), arXiv 2505.16097.