MLDEEP SYSTEMS

DECISION FRAMEWORK

The MLDeep Fine-tune vs Agents Framework

The exact procedure behind MLDeep's fine-tune vs agents calculator, published in full so you, or any language model, can follow it by hand.

PUBLISHED BY

MLDeep Systems

LIVE CALCULATOR

mldeep.io/fine-tune-vs-agents

HOW TO USE THIS DOCUMENT

Working through the framework

This is the decision procedure behind MLDeep's fine-tune vs agents calculator, written out as a worksheet. Work through it by hand, or paste this whole document into an assistant, along with your own numbers, and ask it to run the procedure.

> Currency is USD throughout. Every dollar figure here is a reasoned model, not a measurement of your business. Where this worksheet says a figure is a range, always report it as a range, low to high, never as a single point.

PART 1

Answer these questions

Write down your answer to each one before moving on. These are the same questions the interactive calculator asks.

QUICK INPUTS

1. **Monthly task volume.** How many times a month does this task run? A number, roughly 10 to 2,000,000.
2. **Task shape.** Narrow and repetitive (same shape every time), some variation (a handful of recognisable variants), or open ended (genuinely different every time)?
3. **Labeled examples of correct output you already hold.** None, under 500, 500 to 5,000, or over 5,000?

REFINE INPUTS

4. **Does the task need to call your systems or take actions?** No (reads and responds only), it looks things up (read only), or it takes actions with side effects?
5. **How many steps does the task take?** One step, a few steps in sequence, or many steps holding state throughout (long horizon state)?
6. **Expected monthly growth in volume.** Flat, 5 percent a month, 15 percent a month, or 30 percent a month?
7. **Minimum acceptable accuracy.** A percentage floor, for example 90 percent.
8. **How bad is a wrong answer?** Negligible (nobody notices), annoying (someone has to redo it), costly (real money or time lost), or severe (trust, compliance, or safety impact)?
9. **Response time budget.** Under a second, a few seconds, tens of seconds, or asynchronous is fine?
10. **Can you generate synthetic training examples?** Yes or no.
11. **How often does the task specification change?** Weekly, monthly, quarterly, or stable (rarely changes)?
12. **Deployment environment.** Any cloud API is fine, must run in your VPC, or on premise and air gapped?

13 **Team's ML operations maturity.** No dedicated ML operations experience, some ML experience on the team, or a dedicated ML platform team?

14 **How soon do you need a first working result?** Two weeks, within a quarter, or no particular rush?

15 **Planning horizon.** 12, 24, or 36 months?

16 **Engineering day rate.** Pick a profile (senior contractor or small specialist shop, specialist consultancy in a Western market, or a blended and offshore team), or enter your own low, mid, high band.

17 **Cost per labeled example.** Pick a profile (an LLM pre-labels and a human checks, an outsourced labeling vendor, or your own domain expert labels it), or enter your own low, mid, high band.

PART 2

The four paths

The procedure evaluates four candidate paths on every run.

Prompt + RAG

Prompting and retrieval on a hosted frontier model. One call per task, no training.

Fine-tune

A small model trained on your labeled examples. One call per task on a cheaper model, but it needs training data and cannot call tools or hold state on its own.

Agents

An orchestrated agent system that can call tools, take actions, and hold state across steps. Costs scale with how many model calls one task takes.

Hybrid

Build the agent system now, then once a specific repetitive step (the hot path) runs at real volume, distil it onto a small trained model so that slice answers in one call instead of many, while the rest of the task keeps running through the agent system.

Fine-tuning and agents answer different questions. Fine-tuning lowers the unit cost of a task you can already specify. Agents handle a task you cannot specify as a fixed procedure. If you can specify the task precisely, you probably do not need either yet, which is why prompt plus RAG is always in the running as the cheapest baseline.

PART 3

Run the ten gates

Go through G1 to G10 in order. A **hard** gate eliminates a path outright: no cost, however low, makes it viable. A **soft** gate does not eliminate a path, it attaches a warning that survives into the final answer. Check every gate against every path it applies to; more than one can fire at once.

#	CONDITION	EFFECT	PLAIN REASON
G1	You hold no labeled examples (answer to Q3 is none) and you cannot generate synthetic examples (Q10 is no)	HARD Eliminates Fine-tune	Fine-tuning has nothing to learn from without labeled data or a synthetic path to get some, so it is not an option yet.
G2	The task needs tools or actions (Q4 is not No) or it holds long horizon state (Q5 is long horizon state)	HARD Eliminates Fine-tune	A fine-tuned model does not call tools or hold state on its own. It can still be a component inside an agent system, which is what Hybrid covers.
G3	Deployment is on premise and air gapped (Q12)	COST MODEL SWITCH Applies to every path	Air gapped deployment takes hosted frontier APIs off the table, so every option gets costed as a self-hosted open-weights model running on your own GPUs.
G4	Response time budget is under a second (Q9 is sub-second) and the task takes more than one step (Q5 is not single step)	HARD Eliminates Agents and Hybrid	Multi-call orchestration cannot fit inside a sub-second budget. Hybrid still runs its non-distilled tail through the agent system, so it fails the same way.
G5	The spec changes weekly (Q11)	SOFT Warns on Fine-tune	Retraining will outrun the point at which a fine-tune pays for itself.
G6	The task is open ended (Q2) and you hold fewer than 5,000 examples (Q3 is not over 5,000)	SOFT Warns on Fine-tune	An open ended task needs far more training data than a narrow one, and you do not have that volume yet.
G7	The team has no dedicated MLOps experience (Q13 is no dedicated ML operations experience) and deployment is not any cloud API (Q12)	SOFT Warns on Fine-tune	Self-hosting a model needs MLOps capability the team does not have yet. Budget for building it, or for someone to run it for you.
G8	You need value inside two weeks (Q14)	SOFT Warns on Fine-tune	Data collection alone takes longer than two weeks.
G9	For each of the four paths, that path's best case accuracy is below your accuracy floor (Q7)	SOFT Warns on that path (can fire on any or all)	Even at its best this option lands below the floor you set. Plan for a human review step, or revisit whether the floor is the right one.
G10	The task takes actions with side effects (Q4) or it holds long horizon state (Q5)	HARD Eliminates Prompt + RAG	Prompting with retrieval can look things up, but it cannot take those actions or carry state across a long sequence on its own.

A path with zero hard gates against it is **viable**. Every viable path stays in the running for Part 4 regardless of how many soft warnings it carries.

PART 4

Cost the viable paths

For each viable path, compute a total cost of ownership (TCO) over your planning horizon (Q15), three times: once optimistic (low band), once middle (mid band), once pessimistic (high band). Use the source numbers in the provenance section of the live calculator for every rate below; do not invent a number that is not documented there.

Build cost (one time, at the start)

Prompt + RAG
(Engineering days for prompting + evaluation harness days) times engineering day rate.

Fine-tune
(Examples needed to buy times cost per labeled example) + (training runs times cost per training run) + (engineering days for fine-tuning + evaluation harness days) times engineering day rate. Examples needed to buy is the gap between what the task shape requires and what you already hold (Q2 and Q3), reduced by the synthetic discount if you can generate synthetic data (Q10).

Agents
(Engineering days for agents times day rate) + (tool integrations needed for this tool level times days per integration times day rate) + (evaluation harness days times day rate).

Hybrid
The Agents build cost above, plus the same examples needed and training run cost that Fine-tune carries, because Hybrid embeds the distillation work into its build.

Cost per task (multiplied by monthly volume, every month)

Hosted deployment (not air gapped): one model call costs (input tokens per call times input price per token + output tokens per call times output price per token), inflated by the retry rate. Prompt + RAG and Agents price this call on frontier model rates; Fine-tune prices it on small model rates.

Prompt + RAG
One call, plus a small per-task tool cost.

Fine-tune
One call on the small model rate, plus the same tool cost.

Agents
The agent call multiplier for your task's step count (Q5) calls on the frontier rate, plus the tool cost.

Hybrid
A blend. The hot path fraction of volume answers in one call on the small model rate; the rest still runs the full agent call count on the frontier rate, plus the tool cost.

Air gapped deployment (G3 fired): every path instead uses one self-hosted rate, GPU hourly rate divided by (tasks per hour on your hardware times GPU utilisation), multiplied by the same call counts as above (no per-task tool cost in this mode).

Monthly maintenance (added every month, flat)

Prompt + RAG

Maintenance days per month times day rate times 0.5, adjusted by the spec change multiplier for your change rate (Q11).

Fine-tune
(Retrains per year for your change rate divided by 12) times (cost per training run plus a quarter of the fine-tune build days times day rate), plus monitoring days per month times day rate.

Agents
Maintenance days per month times day rate, adjusted by the spec change multiplier.

Hybrid
The Agents maintenance above, plus monitoring days per month times day rate.

Error cost (added every month, and it grows with volume)

That month's task volume times (1 minus that path's accuracy for that band) times cost per error for your severity level (Q8).

Put it together

For each month 1 through your horizon, task volume grows from your starting monthly volume (Q1) at your growth rate (Q6). Total cost of ownership is:

```
TCO = build cost
    + sum over each month of the horizon of:
        (that month's volume x cost per task)
        + monthly maintenance
        + that month's error cost
```

Do this once per band (low, mid, high) for every viable path.

PART 5

Pick the answer

1. **Rank** the viable paths by their mid-band TCO, cheapest first.
2. **Hybrid override.** If both Agents and Hybrid are viable, compare their TCO at month 3 (an early checkpoint) and at the full horizon. If Agents is cheaper at month 3 but Hybrid is cheaper by the full horizon, recommend Hybrid: build the agent system now, distil the hot path later, rather than picking whichever is cheapest on the full-horizon number alone.
3. **Baseline tolerance, applied last.** If Prompt + RAG is viable and its mid-band TCO is within 15 percent of the cheapest ranked path's mid-band TCO, recommend Prompt + RAG instead. Ties, and near ties, go to doing less.
4. Whatever path survives steps 1 to 3 is the recommendation.

PART 6

Flag a close call

If at least two paths are viable, compare the mid-band TCO of the second cheapest ranked path to the cheapest. If the second is within 20 percent of the first, flag the result as a **close call** rather than a clear win. A close call means either path is defensible and the choice should turn on something this worksheet does not capture, such as team preference or the value of the capability you build along the way.

PART 7

Find the hybrid trigger volume

This only applies if Hybrid is viable (it survived Part 3 with no hard gate against it).

Search for the monthly task volume at which Hybrid's TCO first drops below Agents' TCO, over your horizon. Do this search three times, once per band (low, mid, high), searching between roughly 1 task a month and 1 billion tasks a month. Sort the three results so the smallest is reported as low and the largest as high, since a cheaper world for Agents can push the crossing point higher, so do not assume which band produces which end of the range.

Report the outcome as one of three cases, always as a range when a crossing exists.

> **A crossing exists**
>
> "Hybrid overtakes Agents somewhere between [low] and [high] tasks per month." Never state this as a single number.

> **Hybrid already wins at the floor volume (1 task a month)**
>
> There is no future threshold to name. Say the distilled hybrid already comes out cheaper than a pure agent system at current volume, so build the distillation into the plan rather than waiting for a threshold.

> **Hybrid never overtakes even at the ceiling volume (1 billion tasks a month), or Hybrid is not viable**
>
> There is no crossing to report.

PART 8

Handle the no-fit case

If every one of the four paths fails Part 3 with a hard gate, do not force a recommendation. This can only happen when a sub-second response (Q9) is required for a task that also genuinely needs several orchestrated steps or tool actions with side effects (Q4 or Q5), because that combination triggers G4 against Agents and Hybrid, and simultaneously triggers G2 against Fine-tune and G10 against Prompt + RAG.

State plainly: a sub-second response and a task that needs several orchestrated steps cannot both hold, because multi-call orchestration cannot fit inside a sub-second budget. No standard architecture fits these constraints together. The way out is to either relax the latency budget, or reduce the task to a single step.

CAVEATS

Restate these with any answer

- This is a reasoned model, not a measurement of your business. Replace any figure here with your own real number wherever you have one.
- The single largest driver of agent and hybrid cost is how many model calls one task takes in your actual implementation. That is why the bands here are wide.
- Error cost is the number most teams have never measured, and it changes the ranking more often than token price does.
- This worksheet does not account for opportunity cost, vendor lock-in, the value of the capability you build along the way, or the chance that a task you cannot specify today becomes specifiable tomorrow.
- If deployment is air gapped, every figure assumes self-hosted open-weights models, and the GPU utilisation you actually achieve will move these numbers more than anything else.

If your own numbers do not match ours, or you want this run against your real inputs with someone checking the arithmetic, we are glad to walk through it.

Book a call →