RAG vs Fine-Tuning vs Agents Decision Guide

EG3 · EG3.COM

Reference Card · Updated April 2026

Decision Tree (Text Flowchart)

```
Start: Can better prompting solve this? (test 5-10 variants, chain-of-thought, few-shot)
├── Yes → Stop here. Use advanced prompting.
└── No → Does it need fresh or private external knowledge?
  ├── Yes → Use RAG (or RAG + prompting).
  └── No → Does it need new consistent behavior, style, or reasoning patterns?
  ├── Yes → Consider fine-tuning (if you have 1K+ quality examples).
  └── No → Does it need multi-step tool use, planning, or external actions?
  ├── Yes → Use agents (often on top of RAG or fine-tuned model).
  └── No → You probably don't need any of these. Revisit the problem.
```

Comparison Table: Prompting vs RAG vs Fine-Tuning vs Agents

APPROACH	COST TO START	ONGOING COST	LATENCY	ACCURACY CEILING	SETUP TIME	MAINTENANCE	BEST FOR
Prompting	Near zero	Very low (tokens)	Lowest	Medium (prompt-dependent)	Minutes to hours	Low (prompt updates)	Most tasks, quick prototypes
RAG	Low-moderate (vector DB, embeddings)	Moderate (retrieval + longer context)	Medium	High for knowledge tasks	Days to weeks	Medium (data indexing)	Knowledge bases, docs, fresh data
Fine-Tuning	Moderate-high (data prep + training)	Low (shorter prompts)	Low	High for style/behavior	Weeks	High (retrain on changes)	Domain expertise, consistent tone, fixed tasks
Agents	High	High (multiple calls + tools)	Highest	High for complex workflows	Weeks+	High (tool reliability)	Multi-step reasoning, tool use, planning

Signs You Need RAG (5-6 items)

- Your data changes frequently (prices, policies, news).
- Users ask about specific documents or private knowledge not in training data.
- You must cite sources or avoid hallucinations on facts.
- Dataset is too large to fit in context window.
- Different users or teams need access to different data slices.
- Compliance requires grounding answers in verifiable sources.[1]

Signs You Need Fine-Tuning (5-6 items)

- You need consistent output format, tone, or style every time.
- Task involves specialized reasoning or jargon that prompting fails to teach reliably.
- Domain knowledge is static and well-defined (no frequent updates).
- You have hundreds to thousands of high-quality labeled examples.
- Response must be fast and cheap at high volume.
- Model must internalize rules without retrieving every time.[2]

Signs You Need Agents

- Task requires multiple steps, conditional logic, or backtracking.
- You need to call external tools (search, APIs, databases, calculators).
- Workflow involves planning then execution (research, booking, analysis).
- Single prompt can't reliably complete the full job.
- System must interact with environment or other systems.
- Goal-oriented behavior matters more than single-turn answers.[3]

Cost Comparison Table (Fine-Tuning, Approximate USD as of 2026)

SCALE	OPENAI (E.G. O4-MINI OR SIMILAR)	ANTHROPIC	OPEN SOURCE (LLAMA 3.1 8B CLASS, QLORA ON RENTED GPU)
1K examples	$100 - 500 (training job)	Not offered	$5 - 30
10K examples	$500 - 2,000	Not offered	$20 - 150
100K examples	$3,000 - 10,000+	Not offered	$100 - 800

Note: OpenAI charges training hours plus inference on the fine-tuned model. Anthropic doesn't offer public fine-tuning. Open-source costs are mostly compute (A100/4090 time). Data preparation labor is extra everywhere.[4]

Common Failure Modes

Better Prompting

- Inconsistent outputs across similar queries.
- Fails on edge cases or novel inputs.
- Prompt becomes hundreds of tokens long and expensive.
- Model still hallucinates facts it was never trained on.

RAG

- Poor retrieval (wrong chunks returned).
- Lost in the middle (long context dilution).
- Vector similarity ≠ semantic relevance.
- Outdated or low-quality source documents.

- High token usage from retrieved context.

Fine-Tuning

- Catastrophic forgetting of general capabilities.
- Overfitting to training examples.
- Expensive to update when knowledge changes.
- Requires substantial clean data (harder than it sounds).
- Model loses flexibility on unrelated tasks.

Agents

- Error cascades across steps.
- Tool hallucinations or incorrect API calls.
- Infinite loops or excessive token spend.
- Poor planning on ambiguous goals.
- High latency and cost from multiple model calls.[5]

Start with prompting. Add RAG when knowledge is the problem. Fine-tune when behavior is the problem. Use agents when workflow is the problem. These aren't mutually exclusive. Many production systems combine a fine-tuned model with RAG inside an agent loop. If you're reaching for any of these before exhausting simpler prompts, you're probably solving the wrong problem.