

ADAPTFM WORKSHOP · ICML 2026

AgentRouter: Heterogeneous Model Routing for Cost-Optimal Multi-Step Agentic Workflows

Rudrendu Kumar Paul, Sourav Nandy

Boston University

University of Texas at Austin

Enterprise agents route every call to a frontier model, wasting 60-80% of their inference budget

THE HIDDEN WASTE IN AGENTIC PIPELINES

Existing routing solutions optimize single-turn query assignment, overlooking a property unique to agentic workflows: **subtask complexity varies widely within a single trajectory**.

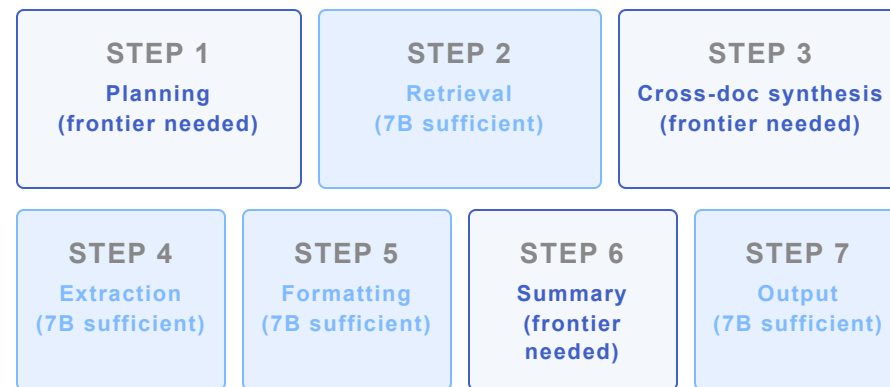
In a 7-step document analysis workflow, only 3 steps need frontier-class reasoning. The other 4 produce identical quality with a 7B model.

Routing all 7 steps to GPT-4-class: \$0.42

Selective routing with AgentRouter: \$0.18 (57% reduction on this workflow), with no measurable quality loss

Illustrative single-workflow example. Benchmark result (72% reduction) is reported across 2,400 held-out trajectories in Section 4.

STEP-LEVEL COMPLEXITY IN A 7-STEP WORKFLOW



☒ Frontier required ☐ 7B model sufficient

AgentRouter: a 12M-parameter classifier that assigns model tier per trajectory step in <5ms

SYSTEM DESIGN

AgentRouter formalizes step-level model routing as a **sequential assignment problem over agent trajectories**. The classifier operates at the boundary of each trajectory step, routing before any LLM call.

12Mclassifier
parameters**<5ms**routing overhead
per step (A100)**50K**annotated trajectory
steps in training set

Training spans planning, coding, research, and data analysis trajectories.

4 MODEL TIERS (ORDERED BY COST & CAPABILITY)

Tier	Profile	Example use
T1 Frontier	Highest capability	Multi-doc synthesis, planning
T2 Mid-range	Strong reasoning	Code generation, analysis
T3 Efficient	Balanced	Summarization, classification
T4 Minimal	7B-class, fast	Retrieval, formatting, extraction

5 FEATURES EXTRACTABLE BEFORE ANY LLM CALL

- Inferred task type: plan, reason, retrieve, generate, format, verify
- Reasoning depth: estimated from instruction complexity
- Tool-use requirements: number of tool calls specified or implied
- Output format constraints: free, structured, code, or JSON
- Context window utilization: ratio of input tokens to model context limit

Single-turn routers fail on trajectories: corrupted intermediate steps degrade every step that follows

THE GAP IN ROUTELLM AND FRUGALGPT

RouteLLM and FrugalGPT treat each LLM call as independent. That assumption holds for single-turn QA. It breaks in agentic workflows, where each step receives the output of the previous step as its input context.

A cheap model producing a subtly wrong intermediate result (a misattributed clause in a contract, a missed entity in an extraction) forces the frontier model at the next step to reason over **corrupted context**. The error propagates forward.

Trajectory-aware routing accounts for **inter-step quality dependencies**. AgentRouter models downstream impact at the moment of routing, before the call is dispatched.

CASCADING QUALITY LOSS: SINGLE-TURN VS. TRAJECTORY ROUTING

SINGLE-TURN ROUTER (ROUTELLM / FRUGALGPT)

Step N: cheap model → subtly wrong output
Step N+1: frontier model receives corrupted context
Step N+2: compound error in synthesis

End-to-end quality degraded; cost still paid

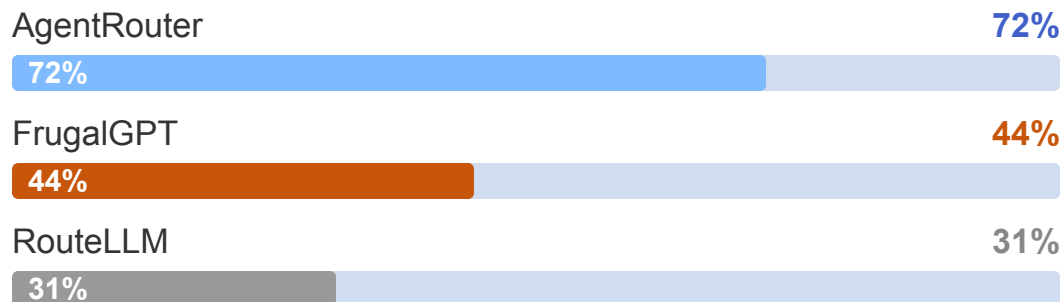
AGENTROUTER: TRAJECTORY-AWARE

Step N: router models downstream deps → correct tier assigned
Step N+1: clean context passed forward

72% cost reduction, <3% quality loss

72% cost reduction vs. frontier-only baselines, outperforming RouteLLM (31%) and FrugalGPT (44%)

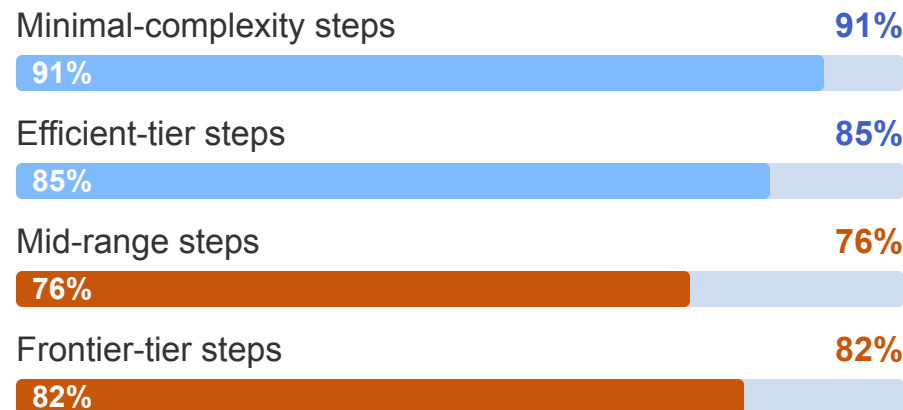
COST REDUCTION VS. FRONTIER-ONLY BASELINE



RouteLLM and FrugalGPT achieve lower reduction because their single-turn training signal misses trajectory-level quality dependencies.

AgentRouter: **97.3% quality retained** (<3% degradation)
in end-to-end task completion vs. frontier-only.

PER-STEP ROUTING ACCURACY BY TIER



Mid-range accuracy is lower by design: over-routing to frontier is the intended failure mode. The classifier errs toward capability when the Tier 1 vs. Tier 2 boundary is ambiguous.

Match model to step complexity: the minority of steps that need a frontier model determines end-to-end quality

CONTRIBUTIONS

- Formalization of **step-level model routing** as a sequential assignment problem over agent trajectories
- AgentRouter: **12M-parameter classifier**, 5 lightweight features extractable at routing time, <5ms overhead per step on A100
- **4-tier model taxonomy** ordered by capability and cost, with per-tier routing accuracy reported
- **72% cost reduction** vs. frontier-only baseline with 97.3% quality retained (<3% end-to-end task completion loss)
- Outperforms RouteLLM (31%) and FrugalGPT (44%): baseline adaptations that treat each call as independent

KEY FINDING

55-70% of agent trajectory steps are satisfiable at sub-frontier capability. The minority that demand frontier reasoning (planning, cross-document synthesis, complex multi-step inference) determines end-to-end quality.

TAKEAWAY

Match model to step complexity. Step complexity determines when frontier reasoning is warranted; task-level assignment wastes it. A frontier model called at the right moment is decisive. Called at every moment, it is just expensive.