PRAXIS: A Production Benchmarking Framework for Multi-Agent AI Systems

Systematic Evaluation of Accuracy-Latency-Cost Tradeoffs in Real-World Deployments

Author: **Sandeep Nutakki**

Affiliation: Independent Researcher

Location: Seattle, Washington, USA

Email: sandeepnutakki@gmail.com

Date: January 2026

Abstract

As multi-agent AI systems transition from research prototypes to production deployments, practitioners face critical decisions about model selection, agent configuration, and resource allocation that directly impact business outcomes. However, existing evaluation approaches focus narrowly on accuracy metrics without considering latency, cost, and scalability constraints that dominate real-world deployments. This paper presents **PRAXIS** (Production Real-world Agent eXecution Intelligence Suite), a systematic benchmarking framework designed to evaluate multi-agent system performance across the accuracy-latency-cost Pareto frontier under realistic production conditions. We introduce standardized task suites spanning reasoning, tool use, and collaboration scenarios, along with measurement methodologies that capture cold-start behavior, streaming latency, and throughput under load. Evaluation of 12 agent configurations across 5 foundation models reveals that optimal configurations vary significantly by task type, with accuracy-latency tradeoffs following predictable patterns amenable to intelligent routing. We demonstrate **production impact** through a Fortune 500 enterprise deployment achieving **68% reduction in operational costs** while maintaining service-level agreements. Our framework enables practitioners to make informed deployment decisions: analysis identifies a **3.2x cost reduction** opportunity through adaptive model selection without sacrificing accuracy. PRAXIS is released as open-source software with an interactive benchmarking dashboard.

Keywords: Multi-Agent Systems, Benchmarking, Large Language Models, Performance Evaluation, Accuracy-Latency Tradeoffs, Production AI, Enterprise Deployment

1. Introduction

The emergence of agentic AI systems—autonomous agents powered by large language models (LLMs) capable of reasoning, planning, and executing complex tasks—has generated significant interest in enterprise applications. These systems promise to automate knowledge work ranging from customer support to data analysis. However, translating research demonstrations into production deployments requires navigating complex tradeoffs between accuracy, latency, cost, and reliability that directly impact business outcomes and user experience.

Current evaluation practices suffer from several limitations that impede production deployment decisions:

- **Accuracy Myopia:** Academic benchmarks prioritize task completion rates without measuring response time or resource consumption—metrics that determine production viability

- **Idealized Conditions:** Evaluations assume unlimited resources, ignoring cold-start delays, rate limits, and concurrent load that characterize real deployments
- **Single-Model Focus:** Comparisons evaluate individual models rather than heterogeneous agent configurations and routing strategies
- **Reproducibility Gaps:** Inconsistent evaluation protocols make cross-study comparisons unreliable, forcing practitioners to re-benchmark for their specific use cases
- **Missing Business Context:** No connection between benchmark results and actual deployment impact (cost savings, SLA compliance, throughput capacity)

This paper addresses these gaps through **PRAXIS**, a comprehensive framework for evaluating multi-agent systems under realistic production constraints. Unlike existing benchmarks that treat evaluation as an academic exercise, PRAXIS is designed to directly inform deployment decisions with quantifiable business impact.

Our contributions are as follows:

1. **Production-Oriented Benchmark Suite:** A standardized benchmark of 500 tasks covering reasoning, tool use, and multi-agent collaboration, designed to reflect enterprise workload distributions with complexity scoring for routing research.

2. **Multi-Dimensional Measurement Framework:** Comprehensive methodologies capturing latency distributions (TTFT, E2E, percentiles), throughput curves under load, cost efficiency, and cold-start behavior—metrics critical for capacity planning and SLA design.

3. **Pareto Frontier Analysis:** Techniques for identifying optimal accuracy-latency-cost operating points, enabling principled configuration selection based on application requirements.

4. **Validated Production Impact:** Empirical demonstration through a Fortune 500 enterprise deployment achieving 68% cost reduction while maintaining accuracy SLAs—bridging the gap between benchmark results and real-world outcomes.

5. **Open-Source Implementation:** Complete framework release including task definitions, measurement harness, analysis tools, and interactive dashboard for reproducible evaluation.

2. Related Work

2.1 LLM Evaluation Benchmarks

Traditional LLM benchmarks focus on capability assessment. MMLU evaluates broad knowledge across academic subjects. HumanEval measures code generation ability. More recently, benchmarks like GAIA and AgentBench target agentic capabilities including tool use and multi-step reasoning.

However, these benchmarks evaluate accuracy under idealized conditions without latency or cost constraints relevant to production deployments. A system achieving 95% accuracy with 10-second latency may be unusable for interactive applications, yet benchmarks treat it equivalently to a 95% accuracy system with sub-second response times.

2.2 Efficiency Evaluation

Prior work has examined LLM efficiency from various angles. FrugalGPT demonstrated cost reduction through model cascading. RouteLLM explored intelligent routing between models of varying capability. These approaches optimize specific metrics but lack comprehensive frameworks for multi-objective evaluation that practitioners can apply directly.

2.3 Production ML Evaluation

The MLPerf benchmark suite established standards for ML system performance evaluation, including latency, throughput, and energy metrics. Our work extends these principles to the unique characteristics of agentic AI systems, including variable-length interactions, tool call dependencies, and the combinatorial space of agent configurations.

2.4 Positioning of PRAXIS

PRAXIS differentiates from prior work through: - **Multi-objective optimization** vs. single-metric focus - **Production conditions** (cold starts, rate limits, concurrent load) vs. idealized evaluation - **Configuration space exploration** vs. single-model comparison - **Business impact validation** vs. abstract metric reporting

3. Framework Design

3.1 Design Goals

PRAXIS is designed around five principles:

1. **Comprehensiveness:** Evaluate accuracy, latency, cost, and reliability jointly with explicit tradeoff analysis
2. **Realism:** Simulate production conditions including cold starts, rate limits, concurrent load, and failure injection
3. **Reproducibility:** Standardized protocols enabling cross-study comparison with published baselines
4. **Extensibility:** Modular architecture supporting new tasks, metrics, and agent configurations
5. **Actionability:** Results directly inform deployment decisions with quantified business impact

3.2 Architecture Overview

Table 1: PRAXIS Architecture Components

Component	Responsibility
Task Suite	Standardized evaluation tasks with complexity scoring
Harness	Execution environment, orchestration, failure injection
Metrics Engine	Multi-dimensional measurement collection and aggregation
Analysis Tools	Pareto optimization, routing simulation, cost modeling
Dashboard	Interactive visualization and comparison interface

3.3 Task Suite

The benchmark includes 500 tasks across three categories, designed to reflect enterprise workload distributions:

Reasoning Tasks (200 tasks)

Multi-step reasoning problems requiring logical inference:

- Mathematical word problems with chain-of-thought reasoning
- Constraint satisfaction and planning puzzles
- Causal reasoning and counterfactual analysis

Complexity distribution: 40% simple (1-2 steps), 35% moderate (3-5 steps), 25% complex (6+ steps)

Tool Use Tasks (200 tasks)

Tasks requiring interaction with external tools:

- Database queries with SQL generation
- API orchestration across multiple services
- File system operations and data transformation

Complexity distribution: 30% single-tool, 45% multi-tool sequential, 25% multi-tool parallel

Collaboration Tasks (100 tasks)

Multi-agent scenarios requiring coordination:

- Debate and consensus-building
- Divide-and-conquer problem decomposition
- Reviewer-author interaction patterns

3.4 Metrics Framework

Accuracy Metrics

- **Task Completion Rate (TCR):** Binary success/failure per task
- **Partial Credit Score (PCS):** Continuous score for partially correct responses
- **Reasoning Quality (RQ):** Human-evaluated explanation quality (1-5 scale)

Reasoning Quality (RQ) was evaluated by two independent annotators blinded to model configuration. Scores were averaged across annotators, with inter-rater agreement of $\kappa = 0.69$ (Cohen's kappa) averaged across task categories, indicating substantial agreement.

Latency Metrics

- **Time to First Token (TTFT):** Initial response latency (critical for perceived responsiveness)
- **End-to-End Latency (E2E):** Total task completion time
- **Percentile Distribution:** P50, P90, P95, P99 latencies (SLA-relevant)

Cost Metrics

- **Token Consumption:** Input and output tokens per task
- **API Cost:** Dollar cost based on provider pricing
- **Cost per Success:** Amortized cost including failures (true deployment cost)

Throughput Metrics

- **Queries per Second (QPS):** Maximum sustainable throughput
- **Concurrent Capacity:** Tasks handled simultaneously without degradation
- **Degradation Curve:** Latency vs. load relationship (capacity planning)

Production Metrics

- **Cold Start Penalty:** Additional latency for first request after idle
- **Error Rate Under Load:** Failure rate at various throughput levels
- **Recovery Time:** Time to return to baseline after saturation

4. Measurement Methodology

4.1 Latency Measurement Protocol

Accurate latency measurement requires careful attention to measurement points:

E2E = T_queue + T_cold + T_inference + T_tool + T_network

where: - T_queue: Time waiting in request queue - T_cold: Cold-start initialization (if applicable) - T_inference: LLM inference time - T_tool: External tool execution time - T_network: Network round-trip overhead

Cold Start Measurement

Cold start behavior is measured by: 1. Allowing the system to idle for 5 minutes 2. Issuing a single request and measuring response time 3. Comparing against warm-state baseline 4. Repeating 10 times for statistical validity

Streaming Latency

For streaming responses, we measure: - TTFT: Time until first token received - Inter-token latency: Average time between tokens - Time to completion: Full response generation time

4.2 Throughput Measurement Protocol

Algorithm 1: Throughput Measurement

INPUT: Target QPS range [q_min, q_max], step size Δq
OUTPUT: Throughput curve C

```
1. C ← []
2. FOR q = q_min to q_max step Δq:
3.     Configure load generator for q QPS
4.     Run for 5 minutes, collecting metrics
5.     metrics ← {latency_p50, latency_p99, error_rate}
6.     C.append((q, metrics))
7.     IF error_rate > 0.05 OR latency_p99 > threshold:
8.         BREAK  // System saturated
9. RETURN C
```

4.3 Statistical Considerations

All measurements are repeated with sufficient samples for statistical validity:

- Minimum 100 samples per task category
- 95% confidence intervals reported for all metrics
- Outlier handling via winsorization at 1st/99th percentiles
- Statistical significance testing (McNemar's test for completion rates)

Differences below 1.5% in completion rate were generally not statistically significant at $p < 0.05$ given our sample sizes, informing the precision of comparative claims throughout this paper.

5. Experimental Evaluation

5.1 Agent Configurations

We evaluated 12 agent configurations spanning 5 foundation models:

Table 2: Evaluated Agent Configurations

Config	Model	Strategy	Cost/1K tokens
A1	GPT-4-Turbo	ReAct	$0.030
A2	GPT-4-Turbo	Plan-Execute	$0.030
A3	GPT-4o	ReAct	$0.015
A4	GPT-4o	Plan-Execute	$0.015
A5	GPT-4o-mini	ReAct	$0.002
A6	GPT-4o-mini	Plan-Execute	$0.002
A7	Claude-3-Opus	ReAct	$0.045
A8	Claude-3-Sonnet	ReAct	$0.009
A9	Claude-3-Haiku	ReAct	$0.001
A10	Llama-3-70B	ReAct	$0.003
A11	Llama-3-8B	ReAct	$0.001
A12	Mixtral-8x7B	ReAct	$0.002

5.2 Accuracy Results

Table 3: Task Completion Rates by Category (%)

Config	Reasoning	Tool Use	Collab.	Overall
A1	91.5	88.0	82.0	88.2
A2	**93.0**	**90.5**	**85.0**	**90.4**
A3	89.5	86.0	79.0	86.0
A4	91.0	88.5	82.0	88.4
A5	78.0	74.5	65.0	74.2
A6	80.5	77.0	68.0	76.6
A7	92.0	89.5	84.0	89.4
A8	86.0	82.5	75.0	82.6
A9	71.5	68.0	58.0	67.4
A10	83.5	79.0	71.0	79.2
A11	62.0	55.5	45.0	56.0
A12	75.0	70.5	62.0	70.6

5.3 Latency Results

Table 4: Latency Distribution by Configuration (seconds)

Config	P50	P90	P95	P99
A1	2.8	5.4	7.2	12.1
A2	4.1	7.8	10.3	16.5
A3	1.9	3.6	4.8	8.2
A4	2.8	5.3	7.0	11.4
A5	0.8	1.5	2.0	3.4

Config	P50	P90	P95	P99
A6	1.2	2.3	3.0	5.1
A7	3.4	6.5	8.6	14.2
A8	1.6	3.1	4.1	6.9
A9	**0.6**	**1.1**	**1.5**	**2.5**
A10	1.4	2.7	3.6	6.0
A11	0.5	0.9	1.2	2.0
A12	0.9	1.7	2.3	3.8

5.4 Cost Efficiency Analysis

Table 5: Cost per Successful Task (USD)

Config	Avg. Tokens	Raw Cost	Cost/Success
A1	3,240	$0.097	$0.110
A2	4,890	$0.147	$0.163
A3	2,980	$0.045	$0.052
A4	4,520	$0.068	$0.077
A5	2,450	$0.005	$0.007
A6	3,780	$0.008	$0.010
A7	3,560	$0.160	$0.179
A8	2,890	$0.026	$0.031
A9	2,120	$0.002	**$0.003**
A10	2,780	$0.008	$0.010
A11	1,950	$0.002	$0.004
A12	2,340	$0.005	$0.007

5.5 Pareto Analysis

Table 6: Pareto-Optimal Configurations

Config	Accuracy	P50 Latency	Cost/Success	Recommended Use Case
A2	90.4%	4.1s	$0.163	Maximum accuracy (complex analysis)
A4	88.4%	2.8s	$0.077	Balanced (general enterprise)
A3	86.0%	1.9s	$0.052	Low latency premium (interactive)
A5	74.2%	0.8s	$0.007	High-volume, cost-sensitive
A9	67.4%	0.6s	$0.003	Minimum latency (real-time)

5.6 Throughput Under Load

Table 7: Throughput Characteristics

Config	Max QPS	Concurrent	Saturation P99
A1	12	35	18.5s
A3	28	55	12.2s

Config	Max QPS	Concurrent	Saturation P99
A5	85	70	5.8s
A9	120	75	4.1s
A11	180	95	3.2s

6. Enterprise Case Study: Financial Services Deployment

To validate PRAXIS beyond synthetic benchmarks, we report results from a production deployment at a Fortune 500 financial services firm (anonymized as "FinCorp"). This case study demonstrates how benchmark insights translate to real-world business impact.

6.1 Deployment Context

FinCorp Background: - Global financial services company with 50,000+ employees - Deployed AI agents for internal knowledge work automation - Initial deployment: Uniform GPT-4-Turbo (A1 equivalent) for all tasks - Monthly volume: ~2.1 million agent invocations - Previous monthly cost: $231,000

Business Requirements: - SLA: 95% of requests completed within 5 seconds - Accuracy: Minimum 85% task completion for business-critical tasks - Cost: Executive mandate to reduce AI operational costs by 40%

6.2 PRAXIS-Guided Optimization

Using PRAXIS analysis, we implemented a three-tier routing strategy:

Table 11: FinCorp Routing Strategy

Task Category	Volume	Selected Config	Rationale
Simple queries	62%	A5 (GPT-4o-mini)	High volume, acceptable accuracy
Standard workflows	28%	A3 (GPT-4o)	Balanced performance
Complex analysis	10%	A2 (GPT-4-Turbo)	Maximum accuracy required

Implementation Details: - Deployed lightweight complexity classifier (fine-tuned DistilBERT, 50ms overhead) - Gradual rollout over 6 weeks with A/B testing - Continuous monitoring via PRAXIS dashboard integration

6.3 Production Results (90-Day Evaluation)

Table 12: FinCorp Deployment Outcomes

Metric	Before	After	Change
Monthly Cost	$231,000	$74,200	**−68%**
Avg. Latency (P50)	2.8s	1.4s	−50%
SLA Compliance	91%	97%	+6 pts
Task Completion	88.2%	86.4%	−1.8 pts
User Satisfaction	3.8/5	4.2/5	+10%

Key Findings:

1. **Cost savings exceeded target:** 68% reduction vs. 40% mandate, saving $1.88M annually

2. **Latency improved dramatically:** Routing simple tasks to faster models improved overall responsiveness

3. **Accuracy tradeoff minimal:** 1.8% completion rate decrease was acceptable given cost/latency gains

4. **User satisfaction increased:** Faster responses improved perceived quality despite slightly lower accuracy

6.4 Lessons from Production

Several insights emerged that extend beyond PRAXIS benchmarks:

1. **Task distribution matters:** FinCorp's 62% simple task ratio was higher than anticipated—accurate workload profiling is essential

2. **Cold starts in practice:** API-based models (A1-A9) showed negligible cold start issues; self-hosted alternatives would have required keep-warm infrastructure

3. **Failure cascade risk:** Circuit breaker patterns from Nexus architecture prevented cost spikes during model provider outages

4. **Monitoring is essential:** Real-time PRAXIS dashboard enabled rapid response to accuracy degradation events

7. Analysis and Insights

7.1 Accuracy-Latency Tradeoff Patterns

Our analysis reveals consistent patterns across configurations:

$$\text{Accuracy} \approx \alpha \cdot \log(\text{Latency}) + \beta$$

where α and β are model-family dependent. Log-linear regression achieved R^2 values between 0.62–0.78 across model families, confirming the logarithmic relationship. This suggests diminishing returns: doubling latency budget yields approximately 5-8% accuracy improvement.

Implication: Beyond a threshold (approximately 3s for current models), additional latency budget provides minimal accuracy gains—useful for SLA design.

7.2 Task-Specific Optimal Routing

Table 8: Optimal Configuration by Task Type

Task Type	Optimal	Runner-up	Routing Signal
Simple reasoning	A5 (74%, 0.8s)	A9 (67%, 0.6s)	Low complexity score
Complex reasoning	A2 (93%, 4.1s)	A7 (92%, 3.4s)	High complexity score
Single tool call	A5 (75%, 0.8s)	A8 (83%, 1.6s)	Tool count = 1
Multi-tool orchestration	A4 (89%, 2.8s)	A1 (88%, 2.8s)	Tool count > 2
Two-agent collaboration	A8 (75%, 1.6s)	A3 (79%, 1.9s)	Agent count = 2
Multi-agent coordination	A2 (85%, 4.1s)	A7 (84%, 3.4s)	Agent count > 2

7.3 Cost Optimization Opportunity

Table 9: Cost Optimization via Intelligent Routing

Strategy	Accuracy	Avg. Cost/Task	Annual Cost (1M tasks)
Always A1 (GPT-4-Turbo)	88.2%	$0.110	$110,000
Always A5 (GPT-4o-mini)	74.2%	$0.007	$7,000
Intelligent Routing	**86.8%**	**$0.034**	**$34,000**
Savings vs. A1	−1.4%	**3.2x**	**$76,000/year**

7.4 Cold Start Impact

Table 10: Cold Start Latency Overhead

Config	Cold Start	Overhead vs. Warm
A1	4.2s	+50%
A5	1.8s	+125%
A10	8.5s	+507%
A11	3.2s	+540%

Self-hosted models (A10, A11) exhibit the largest cold start penalties, making them less suitable for sporadic workloads without keep-warm infrastructure investment.

Deployment Guidance: In practice, these penalties suggest that self-hosted models are best suited for sustained, high-throughput workloads where cold starts are amortized across many requests, while API-based models are more economical for bursty or unpredictable traffic patterns.

8. Discussion

8.1 Implications for Practitioners

Our findings suggest several actionable deployment strategies:

1. **Implement Task Routing:** Use lightweight classifiers to route tasks to appropriate configurations—PRAXIS provides training data and baseline accuracy

2. **Set Latency Budgets:** Choose configurations based on application latency requirements; the logarithmic accuracy-latency relationship enables principled tradeoffs

3. **Monitor Cost per Success:** Raw API costs understate true costs when including failures and retries; PRAXIS cost metrics reflect production reality

4. **Consider Cold Start:** Serverless architectures may require keep-warm strategies for self-hosted models; API-based models generally have acceptable cold start behavior

5. **Profile Your Workload:** The FinCorp case study demonstrates that workload distribution significantly impacts optimal configuration; PRAXIS dashboard helps characterize task complexity distribution

8.2 Architectural Implications for Agent Design

Beyond configuration selection, PRAXIS results inform fundamental agent architecture decisions:

- **Planner Depth vs. Latency:** Plan-and-Execute strategies (A2, A4, A6) consistently outperform ReAct on complex tasks but add 40-60% latency overhead. Use Plan-and-Execute when task complexity exceeds 4 steps; default to ReAct otherwise.

- **When Multi-Agent Adds Value:** Two-agent configurations showed diminishing returns beyond debate/review patterns. Multi-agent coordination (3+ agents) is only justified for tasks requiring genuine specialization—not as a general accuracy boost.

- **Collapsing Agent Flows:** Many production deployments over-engineer agent interactions. Our data suggests that 62% of enterprise tasks (per FinCorp) are handled optimally by single-turn or simple ReAct loops. Reserve complex orchestration for the tail.

- **Model Heterogeneity in Pipelines:** Consider using fast/cheap models (A5, A9) for initial triage and routing decisions, reserving expensive models (A1, A2, A7) for final synthesis. This "funnel" pattern underlies the 3.2x cost savings.

8.3 Limitations

Several limitations should be acknowledged:

1. **Model Evolution:** Results reflect specific model versions (January 2026); capabilities and pricing evolve rapidly

2. **Task Coverage:** Our benchmark may not represent all production use cases; organizations should supplement with domain-specific evaluation

3. **Provider Variability:** Latency varies by time of day, region, and provider load; PRAXIS measurements represent controlled conditions

4. **Cost Volatility:** API pricing changes frequently; cost projections require periodic recalibration

8.4 Reproducibility and Open Science

To support reproducibility and community advancement, we provide:

- **Complete task definitions:** 500 tasks with ground truth and complexity labels
- **Measurement harness:** Standardized protocols for latency, throughput, and cost measurement
- **Analysis tools:** Pareto optimization, routing simulation, and cost modeling
- **Raw results:** All configuration measurements for independent verification
- **Interactive dashboard:** Web-based comparison and visualization tool

Repository: https://github.com/sandeepnutakki/praxis-bench

9. Conclusion

This paper presented **PRAXIS**, a systematic framework for evaluating multi-agent AI systems across accuracy, latency, cost, and throughput dimensions under realistic production conditions. Our evaluation of 12 agent configurations revealed significant variation in optimal configurations by task type, with a **3.2x cost reduction** achievable through intelligent routing without meaningful accuracy sacrifice.

Key findings include:

1. **Accuracy-latency tradeoffs follow logarithmic patterns** with diminishing returns beyond ~3s latency budget
2. **Optimal configuration varies dramatically by task complexity**, motivating intelligent routing as a first-class deployment concern
3. **Cold start penalties significantly impact self-hosted deployment economics**, favoring API-based models for variable workloads
4. **Cost per successful task** is the appropriate metric for production planning, often 10-20% higher than raw API costs

Production validation through the FinCorp case study demonstrates that PRAXIS insights translate directly to business impact: **68% cost reduction** ($1.88M annual savings) while improving latency SLA compliance from 91% to 97%.

We release PRAXIS as open-source software to enable practitioners to make informed deployment decisions and to advance community understanding of production agentic AI system performance. Results are representative of current model generations and pricing as of January 2026; we commit to maintaining updated benchmarks as the landscape evolves.

9.1 Companion Materials

To maximize impact beyond the academic paper, PRAXIS includes:

- **GitHub Repository:** Complete source code, task definitions, and measurement infrastructure (https://github.com/sandeepnutakki/praxis-bench)
- **Interactive Dashboard:** Web-based benchmarking results viewer with custom comparison tools
- **Practitioner Guide:** Medium/Substack explainer series covering deployment best practices
- **Benchmark Results API:** Programmatic access to all measurement data for integration into MLOps pipelines

9.2 Future Work

Future research directions include:

- Continuous benchmark updates tracking model evolution
- Domain-specific task suites (legal, medical, financial)
- Multi-provider routing strategies for reliability
- Energy efficiency metrics for sustainability reporting
- Integration with CI/CD pipelines for continuous evaluation

Acknowledgment

The author thanks the reviewers for their feedback, the FinCorp team for production deployment collaboration, and the open-source community for evaluation infrastructure.

References

1. S. Yao et al., "ReAct: Synergizing Reasoning and Acting in Language Models," in *Proc. ICLR*, 2023.

2. L. Wang et al., "A Survey on Large Language Model based Autonomous Agents," arXiv:2308.11432, 2023.

3. D. Hendrycks et al., "Measuring Massive Multitask Language Understanding," in *Proc. ICLR*, 2021.

4. M. Chen et al., "Evaluating Large Language Models Trained on Code," arXiv:2107.03374, 2021.

5. G. Mialon et al., "GAIA: A Benchmark for General AI Assistants," arXiv:2311.12983, 2023.

6. X. Liu et al., "AgentBench: Evaluating LLMs as Agents," in *Proc. ICLR*, 2024.

7. L. Chen et al., "FrugalGPT: How to Use Large Language Models While Reducing Cost," arXiv:2305.05176, 2023.

8. I. Ong et al., "RouteLLM: Learning to Route LLMs with Preference Data," arXiv:2406.18665, 2024.

9. P. Mattson et al., "MLPerf Training Benchmark," in *Proc. MLSys*, 2020.

10. OpenAI, "GPT-4 Technical Report," arXiv:2303.08774, 2023.

11. Anthropic, "Claude 3 Model Card," 2024.

12. H. Touvron et al., "Llama 2: Open Foundation and Fine-Tuned Chat Models," arXiv:2307.09288, 2023.

13. A. Jiang et al., "Mixtral of Experts," arXiv:2401.04088, 2024.

14. J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," in *Proc. NeurIPS*, 2022.

15. T. Brown et al., "Language models are few-shot learners," in *Proc. NeurIPS*, 2020.