

CLEA WORKSHOP · NEURIPS 2026 · OPINION PAPERS TRACK

Position: The MLOps-to-LLMOps Transition Demands a New Operational Paradigm

Rudrendu Kumar Paul Sourav Nandy

Boston University

University of Texas at Austin

Enterprises with mature MLOps face a concrete question: which practices still work for generative AI?

A COMPOSITE SCENARIO

A retrieval-augmented generation pipeline passes every CI/CD test and meets its latency SLA. Then answer-quality complaints climb. The retrieval index drifted as source documents changed, and the MLOps monitoring stack, built around latency percentiles and error rates, reports nothing.

Every request still returns a well-formed response within budget, so no infrastructure-level signal fires while answer quality erodes.

FOUR ARCHITECTURAL DEMANDS OF GENERATIVE AI

- Natural language as a control surface
- Non-deterministic outputs
- Unbounded context
- Autonomous multi-step execution

Prior work defines MLOps architecture and proposes LLMOps lifecycle frameworks, but no systematic mapping between the two regimes exists. This paper supplies one.

One architectural test sorts 23 operational capabilities into three transfer classes

THE TEST

For each capability, does generative AI change the **failure surface**, the **verification method**, or the **control interface**, relative to predictive ML? We coded all 23 capabilities ourselves against the cited literature, as a single rater.

Borderline calls. Alerting changes its verification method (semantic drift replaces a fixed threshold) but keeps a recognizable counterpart: adapt. Model routing changes the control interface itself: new.

Sanity check: public documentation of nine LLM application platforms, from LangSmith to Portkey.

THE 11-7-5 CLASSIFICATION

11

Transfer directly. Nothing changes: CI/CD, version control, incident response, access control, and more.

7

Require adaptation. Something changes, but an MLOps counterpart exists: testing, alerting, feature stores.

5

Are new. No predictive-ML analog exists.

Seven practices keep their function, but their verification logic must be rebuilt

Practice	MLOps form	LLMOps change
Testing strategies	Accuracy, precision, recall on held-out sets	Multi-dimensional rubrics: correctness, groundedness, safety, format
Deployment patterns	Blue-green with accuracy gate	Shadow-scoring with human-in-the-loop review
Feature stores	Numerical feature vectors	Versioned prompt template registries
Experiment tracking	Hyperparameter sweeps, loss curves	Prompt variant comparison, chain config diffs
Data pipelines	Structured ETL, feature engineering	Unstructured ingestion, chunking, embedding, indexing
Model registries	Model artifact plus metadata	Compound configs: model, prompt, chain, tools
Alerting	Threshold-based on prediction metrics	LLM-judged quality monitoring

Testing is the most consequential shift. LLM tests must check that output is correct, grounded, free of hallucinated claims, formatted, and safe.

Five capabilities have no MLOps precedent

THE FIVE NEW CAPABILITIES

- **Prompt lifecycle management:** natural language had no role as a control surface
- **Context optimization:** fixed-schema ML carries no token budget
- **Guardrail engineering:** predictive outputs are constrained by design
- **Model routing:** one model per endpoint leaves no routing decision
- **Agent orchestration:** autonomous multi-step execution has no analog

WHY THEY MATTER FOR ENTERPRISE AGENTS

As enterprises move to multi-agent systems that keep adapting in production, these five capabilities form the operational substrate that continual adaptation depends on.

Model routing. RouteLLM (Ong et al. 2024) achieves over 2x cost reduction by dynamically selecting between stronger and weaker models. Whether a learned router beats hand-coded rules remains open.

Agent orchestration. The least mature capability. Practice relies on hand-coded control logic; a learned policy is unexplored.

The adapt category is the bottleneck, and the 11-7-5 split is a hypothesis awaiting a fielded study

A TESTABLE HYPOTHESIS

We hypothesize, but have not measured, that organizations further along the 23-capability checklist deploy LLM features faster and with fewer incidents.

- Survey practitioners at enterprises running both predictive and generative AI
- Control for enterprise size and model family, with stratified sampling
- The split fails if respondents with only the eleven direct-transfer practices look the same

HONEST LIMITATIONS

- **Position paper:** no fielded study or primary practitioner data
- **Single rater:** we coded the classification ourselves
- **Platform review:** convergence is correlational evidence and does not prove need
- **Generality:** may not hold across scale, industry, or maturity

Retrieval drift raises no alert. A connection-pool exhaustion pages an engineer within minutes. MLOps tooling was built to catch the second.

Carry forward what transfers, rebuild what adapts, invest early in what is new

KEY TAKEAWAYS

1. LLMOps is a distinct operational regime that goes beyond incremental MLOps.
2. The seven adaptation practices are the blind spot: rebuild testing and monitoring for non-deterministic behavior.
3. Treat 11-7-5 as a checklist awaiting validation, and invest early in the five capabilities with no precedent.

Scan for paper on OpenReview

<https://openreview.net/forum?id=Uvjw8BuVBA>

