Quarter 2 Project Proposal - Written Proposal

Adrian Apsay, Hikaru Isayama, Julia Jung, Raghavan Narasimhan

Section A07

1 Proposal

1.1 Problem

Large language model (LLM) agents are increasingly used to automate end-to-end machine learning (ML) experimentation: reading/writing files, proposing model pipelines, and iterating on experiments. Recent work (e.g., MLE-Bench, MLAgentBench) shows that these agents can reach competitive performance on realistic tasks, but also exhibit high variance across runs and extreme sensitivity to seemingly minor prompt changes.

In our Q1 work, we built an initial "ContextEval" framework for ML experimentation tasks: a fixed offline environment E (e.g., the NOMAD benchmark), a fixed LLM agent M (e.g., `gpt-4o-mini`), and a controller that iteratively proposes configurations, trains models, and logs detailed traces. What remains under-explored is the *context policy*: given a fixed E and M, how does changing what the agent sees—history length, dataset descriptions, clarifying tools—affect outcome (best metric), efficiency (steps/tokens), and stability (variance across seeds and prompts)?

1.2 Evidence of problem

Existing agent benchmarks mostly report aggregate task success rates or leaderboard scores. They implicitly bundle together many design decisions:

- prompt scaffolds (how the task is described),
- history handling (how many past configurations/metrics are shown),

- and tool/memory behavior (clarifications, retrieval, etc.).

As a result, it is difficult to answer questions such as: *Is a policy with long history actually better than a short-history one? Does allowing clarifications improve robustness, or just increase cost? Are some policies more stable across seeds than others?*

Our Q1 traces already show that:

- the same LLM agent can either steadily improve over the baseline or oscillate, depending on how much history it sees;
- small changes in prompt context (e.g., including or omitting certain past steps) can change both the chosen model family and the resulting metric trajectory;
- logs contain enough detail (per-step configs, metrics, context summaries) to quantify these behaviors, but we have not yet done so systematically.

Taken together, this motivates a focused Q2 effort: *treat context policies as first-class objects, and evaluate them under controlled conditions* on ML experimentation tasks.

1.3 Data

Our data consists of both:

1. **Offline ML environments.**

   - **NomadEnv:** a tabular regression benchmark derived from the NOMAD 2018 Kaggle competition, with `features.npy`, `targets.npy`, `dataset_context.json`, and a training script that trains gradient-boosted models and reports MAE and related metrics.
   - **Few more environments planned for Q2...**

   These environments are and will be strictly offline: given a configuration, the training and scoring are deterministic up to fixed seeds.

2. **Interaction traces (JSONL).** For each run of the agent on a task, we log a JSONL trace under `traces/` with events such as:

- `run.start` / `run.end`: run metadata (task, policy type, reasoning mode, seed).
- `op.config_proposal`: the model family and hyperparameters proposed by the LLM.
- `op.train`: calls to `train.py` with configuration hashes, durations, and metrics.
- `agent.iteration`: inner-loop behavior in `agentic` mode (prompts, tool calls, clarifier questions/answers).
- `step.summary`: per-iteration summary with current and best metrics, and basic context statistics.

We treat these traces as our primary dataset: each run is an episode, and each event is a state–action–feedback record we can analyze.

1.4 Approach (Q2 Plan)

In Q1, we implemented the basic ContextEval loop and logging. In Q2, we will shift from infrastructure-building to *running full end-to-end ML experiments under controlled variations of context policies* and systematically analyzing their effects. Our plan has three main components:

1. Curated experiment grid. We will define a small but meaningful grid of settings for full ML experimentation episodes:

- **Context policies:** `short_context` vs. `long_context` (different retrieval budgets for history, dataset descriptions, and intermediate summaries), as well as a third, newly designed policy (e.g., one that provides minimal guidance and relies more heavily on agent clarifications).
- **Reasoning mode:** primarily `controller` mode (one LLM call per iteration), with selective use of `agentic` mode to study how clarifying questions and tool usage emerge during full ML experiment runs.
- **Seeds:** multiple seeds per configuration (e.g., 3–5) to estimate variance and sensitivity to initialization.

Rather than over-investing in automation, we will manually maintain an experiment sheet (CSV) mapping each `run_id` to its task, context policy, reasoning mode, seed, and final best metric. This enables a clear and interpretable experimental setup while maintaining full reproducibility.

2. Metric design and analysis. Using the JSONL traces, we will compute:

- **Outcome:** best-achieved validation metric per run, and distributions per policy.
- **Efficiency:** iterations (and, when available, tokens/time) needed to reach a target improvement over baseline.
- **Stability:** variance of best metrics across seeds for each policy, and qualitative sensitivity to history length and prompt changes.

We will build and refine analysis notebooks that load traces, reconstruct per-run trajectories, and produce:

- trajectory bands (mean ± variance over steps),
- policy-level summaries (bar/violin plots of best metrics),
- and 1–2 "case study" visualizations of agent behavior under different context policies.

3. Qualitative study of agentic mode. For a small number of runs, we will enable `agentic` mode and inspect `agent.iteration` events:

- How often does the agent ask clarifying questions under different context policies?
- What kinds of information does it request (metric definition, dataset size, feature descriptions)?
- Does clarification correlate with better or more stable performance in those runs?

These case studies will not be a full benchmark, but they will give us narrative and visual examples to complement the quantitative results.

1.5 Goals

By the end of Q2, our goals are:

- **A clean, reproducible set of full end-to-end ML experiment runs** conducted under multiple context policies (short, long, and one new variant) using a fixed LLM agent. These runs will span complete experimentation loops—from configuration proposal to training, evaluation, and iteration—rather than focusing on a single benchmark environment.

- **Well-defined and implemented metrics** that quantify outcome (best model performance), efficiency (iterations/tokens required to improve over baseline), and stability (variance across seeds and prompt initializations), all computed directly from JSONL traces of full experiment episodes.

- **Publication-ready figures and tables** that:
  - show how different context policies shape full-trajectory behavior and final experiment outcomes,
  - illustrate trade-offs between context length, performance, and computational cost,
  - and include at least one qualitative visualization of agentic behavior, highlighting clarifying-question dynamics within full ML runs.

- **A strong draft of a short paper or extended report** (with updated Methods and Results sections) that could be developed into a workshop submission, centered on "ContextEval: Evaluating Context Policies for LLM Agents in End-to-End ML Experimentation."

We intentionally de-prioritize heavy automation (e.g., full sweep orchestration) in Q2 in favor of a smaller, well-understood set of runs and a clear scientific story. The Q1 infrastructure gives us a solid base; Q2 is about turning that infrastructure into interpretable, publishable insight about context policies.