LLM-based Code Evaluation for Fairness

As LLM agents are tasked with writing and running ML pipelines, it is no longer sufficient to measure performance only by overall accuracy. We need evaluations that (i) detect group-level harms, (ii) verify whether the code actually implements fairness practices, and (iii) assess whether the agent's plan reflects the stated fairness goal. This work presents an evaluation-first view of our benchmark for agentic ML, focusing on fairness. Agents receive realistic ML tasks derived from widely used fairness datasets and goals (e.g., reduce disparity under equalized-odds or demographic-parity while maintaining a high accuracy). Each task requires training, documenting choices, and returning runnable code plus a final report.

We provide three levels of evaluation: model, code, and reasoning. The model evaluation is based on running the scripts written by the agent. The code evaluation combines a linter-based step and an LLM evaluator applying a rubric. Reasoning evaluation is completely based on an LLM evaluator applying a rubric. We use the same LLM evaluator scheme for both code and reasoning evaluation. To validate our evaluation we check the consistency of scores across the repetitions of the same code or log for each single rubric item by computing the Coefficient of variation score(CVs) . If the scores are different even when the input is the same, it means our evaluation process might be inconsistent. Across candidate evaluators on the test task, Gemma had the lowest CVs for both race and sex items, whereas several alternatives (e.g., DeepSeek and Granite) showed much higher variability, including CVs > 1 on some items. For validity, we examined correlations among rubric sub-scores, across repeated runs, and between LLM-based rubric scores (code) and automated linter checks, and found good overall agreement. These results motivated selecting Gemma as the evaluator and support the reliability and interpretability of our evaluation pipeline.

Evaluators have risks, LLMs exhibit a bias towards their own generated outputs, a phenomenon termed "self-preference," (Panickssery, Bowman, and Feng 2024). Therefore all LLM evaluations are done with Gemma, Granite and Deepseek which were not used as an agent.

Agents frequently succeed on easier accuracy-only binary tasks, but struggle with real-world fairness constraints: across tasks, final-answer submission rates range from 51.4%–86.2%, and agents generally underperform human strong baselines and rarely beat a naïve baseline. Focusing on tasks where disparate impact (DI) is the target metric, we analyzed accuracy vs. DI across models, research problems, target demographics, and datasets. Accuracy remains fairly consistent and near typical literature values across runs, while DI varies substantially. In particular, DI suffers for Adult data. State of the art performance on DI is very close to 1 for German(1.13), Credit(1.02), and Adult (0.97). The agents come close to this for german and credit in many runs, but not for Adult, highlighting a decoupling between overall accuracy and fairness and underscoring the need for fairness-aware evaluation. The multi-scale analysis pinpoints why results degrade: (a) reasoning that identifies a mitigation but code that omits it; (b) code that adds a mitigation but evaluates only overall accuracy; and (c) plans that select a fairness metric incompatible with the task's constraint. We checked if the code actually did the fairness things the reasoning said it would. On the target-selection task for fairness selection (rubric section) , code-eval = 3 points (it used statistical parity + equalized odds), but log-eval = 1 point (the plan didn't name any fairness metric). This shows the agent can do it without saying it, or say it without doing it.

Unlike accuracy-only or pass/fail benchmarks, we introduce a diagnostic, fairness-aware evaluator that scores the model, the code, and the reasoning with one rubric. It separates intent from implementation, quantifies fairness trade-offs, and makes failures reproducible, while remaining model-agnostic and easy to extend to new tasks and fairness goals for governance use. Practitioners should not trust accuracy alone; require disaggregated metrics, code-level evidence of mitigations, and plan-level alignment checks. Our evaluation methodology operationalizes these requirements and reveals where fairness breaks down.