

Beyond Leaderboard Benchmaxing: Using Adversarial Validation and Counterfactual Probes to Measure Real User Experience

Taylor Amarel · Kaggle Grandmaster (#9 Competitions / #6 Notebooks) · Portal: <https://papers.baltor.ai> · Blog: <https://blog.baltor.ai>

Core Argument: In competitive machine learning on Kaggle, when your validation score looks great but your live test score collapses, the first tool you reach for is **Adversarial Validation (AV)**—training a classifier to tell whether a sample came from the benchmark or from real production traffic. Today’s LLM leaderboards suffer from severe distribution shift and benchmaxing: models score 90%+ on static benchmarks yet fail on routine brownfield repositories, truncated JSON streams, or camouflaged safety traps. This paper introduces an Adversarial Validation and Real-User-Experience (RUX) framework for evaluating models and harnesses honestly.

1. EIGHT WAYS LLM BENCHMARKS GET GAMED (“BENCHMAXING”)

Pathology	How It Inflates the Leaderboard Score	What Actually Happens to a Real User
1. Paraphrase & Constant Memorization	Model memorizes canonical GSM8K / HumanEval phrasing and numbers	Renaming variables or shifting constants causes a 20%–35% accuracy drop
2. Judge Sycophancy & Markdown Padding	Writing 3× longer lists and flattering openers wins LLM-as-a-Judge arenas	Wastes developer reading time and buries subtle bugs in verbose prose
3. Disguised pass@100 Sampling	Sampling 100 trajectories over 40 minutes to claim a headline SOTA percentage	Unusable in interactive coding where a developer needs pass@1 in under 10 seconds
4. Needle-in-a-Haystack Illusion	100% lexical retrieval of a random sentence at 128K context length	Multi-hop reasoning across 3 dispersed clauses degrades sharply past 24K tokens
5. Safety Over-Refusal Theater	Refusing any prompt containing sensitive legal, security, or medical words	Blocks legitimate NGO caseworkers, security auditors, and honest employers

2. MEASURING REAL USER EXPERIENCE (‘RUX’) ACROSS HARDWARE & CODING RESETS

Instead of relying on a single static benchmark number, our evaluation harness in Baltor tracks four ground-truth production signals:

- Adversarial Validation AUC (AV_{AUC}):** Measures how easily a classifier separates benchmark prompts from real brownfield user sessions ($AV_{AUC} \approx 0.50$ means realistic; $AV_{AUC} > 0.80$ flags a synthetic toy benchmark).
- Counterfactual Perturbation Stability:** Every benchmark item is paired with a symbol-renamed, constant-shifted, and benign-twin counterfactual.
- Coding Reset Rate (‘git reset’ / Immediate Revert Telemetry):** Tracks how often a developer or downstream verifier has to revert or manually rewrite an agent’s output within 5 minutes.
- Hardware-Grounded Fit (‘HW1..HW6’):** Evaluates whether a workflow actually runs within a user’s real memory and latency envelope (from a 16GB RAM laptop or single 24GB local GPU to Ollama Cloud workhorse models like Gemma 4 and Qwen3-Coder).