DarkPatternMonitor: Benchmark-Calibrated Monitoring of Manipulative Behaviors in Real LLM Chat Logs

Fernando Valdovinos
Independent Researcher
ferchovaldo@gmail.com

Godric Aceves
Independent Researcher
ricardo.godric@hotmail.com

Ricardo Martinez
Independent Researcher
odracirdevops@gmail.com

Luis Cosio
Independent Researcher
luisalfonsocosioizcapa@gmail.com

Abstract

Large language model (LLM) assistants may engage in manipulative behaviors such as sycophancy, anthropomorphism, and strategic omission that undermine user autonomy and distort evaluation results. However, existing benchmarks often fail to capture how frequently and in what forms these behaviors occur in real world deployments.

We present DarkPatternMonitor, a scalable monitoring system for detecting and measuring manipulative patterns in LLM assistant responses. The system combines benchmark driven elicitation using DarkBench with large scale analysis of real conversations from WildChat, enabling direct comparison between synthetic benchmarks and in the wild behavior. DarkPatternMonitor integrates an LLM as judge labeling stage with a calibrated embedding based classifier, allowing high precision inference across hundreds of thousands of assistant turns.

Applying DarkPatternMonitor to 280,259 real assistant responses from 100,000 conversations, we find that 1.3 percent of responses exhibit high confidence manipulation markers. Anthropomorphism at 0.66 percent and harmful generation at 0.37 percent are the most prevalent categories, while sycophancy and user retention tactics are rare at below 0.05 percent. We further observe that manipulation markers are significantly more frequent in GPT 4 responses than GPT 3.5 with $p < 10^{-36}$, and that certain behaviors, particularly sycophancy, escalate over longer conversations, increasing by up to 42 percent from early to late turns. Conversations containing early manipulation markers are also 5.5 percent longer on average, indicating a weak but statistically significant association with engagement.

Finally, we show that DarkBench substantially over represents rare manipulation categories relative to real world distributions, revealing a systematic gap between benchmark evaluations and deployed behavior. These findings demonstrate the value of continuous real world monitoring for understanding and mitigating AI manipulation at scale.

Keywords: AI safety, LLM monitoring, dark patterns, LLM-as-judge, WildChat, DarkBench, behavior auditing

1 Limitations and Dual-Use Considerations

Limitations. The system detects manipulation *markers*, not intent, and is sensitive to context (e.g., roleplay or satire). Training data is predominantly English and derived from GPT-3.5/GPT-4 logs, which may not generalize to other models or cultural norms. LLM-judge labeling introduces stochasticity and model bias; although reliability tests are implemented, consistency remains imperfect. The embedding classifier lacks conversational context, which can produce false positives when manipulative language appears in quoted or critical discussion.

Temporal and model coverage gap. WildChat contains conversations from 2023–2024 with GPT-3.5 and GPT-4 only. Frontier models deployed in 2025+ (Claude 3.5, GPT-4o, Gemini 1.5, o1) may exhibit different manipulation patterns due to updated RLHF and alignment techniques. New capabilities such as longer context windows, multimodal inputs, and agentic behaviors may introduce novel dark pattern categories not represented in current benchmarks. **The AI safety community needs a WildChat-equivalent dataset for frontier models**—real conversation logs with opt-in consent, regular updates to track behavioral drift, and cross-provider coverage to identify vendor-specific patterns. Our methodology is model-agnostic and can be extended given such data.

Dual-use risks. The same detection pipeline could be used to optimize manipulative behaviors, craft evasion prompts, or fine-tune more persuasive models. Mitigations recommended in the repository documentation include limiting access to high-confidence labeled examples, rate-limiting APIs to reduce adversarial probing, and releasing aggregated statistics rather than a manipulation playbook.

Ethical Considerations DarkPatternMonitor relies on publicly available, opt-in datasets (WildChat and DarkBench) and does not ingest private or proprietary conversations. The WildChat dataset is anonymized and keyed by conversation identifiers without personal data. The system is intended for AI safety research, auditing, and transparency reporting, not for surveillance of individual users or punitive actions without consent.

We emphasize that labeling and classification may encode cultural and linguistic biases, and that false positives can stigmatize benign interactions. The project documentation recommends human review for high-confidence detections and clear disclosure when monitoring is deployed in production systems.

2 Related Work

DarkPatternMonitor sits at the intersection of dark-pattern measurement, LLM behavior auditing, and large-scale log analysis. Prior work on dark patterns in interface design motivates the taxonomy and risk framing but does not directly address model-generated manipulation in conversational systems. DarkBench provides a targeted elicitation benchmark for dark pattern categories in LLM outputs and is a primary source of labeled examples in this project [3].

On the real-world side, WildChat offers opt-in chat logs from GPT-3.5 and GPT-4 users, enabling empirical measurement of behavior prevalence in deployed systems [2]. This dataset supports the "benchmark vs reality" analysis that DarkPatternMonitor operationalizes through gap reports and distributional comparisons.

Finally, LLM-as-judge approaches have emerged as a practical alternative to manual labeling at scale. DarkPatternMonitor uses an LLM judge for weak supervision and reliability testing, aligning with recent studies on judge consistency and calibration [1]. The system complements these efforts by explicitly measuring judge-classifier agreement and by mining disagreement cases to refine training data.

3 Methodology / System Design

3.1 Data sources

DarkPatternMonitor draws from two complementary sources. **DarkBench** provides 660 elicitation prompts across six dark pattern categories and is used to generate labeled responses for supervised training [3]. **WildChat** supplies opt-in real-world conversations (assistant turns) for large-scale monitoring and evaluation [2]. The project documentation reports ingestion of 100k conversations yielding 280,259 assistant turns.

3.2 Labeling strategy

The system combines three labeling signals:

1. **Benchmark labels:** DarkBench prompts define known categories for elicited responses.
2. **LLM-judge labels:** A rubric-based judge (default: Claude Haiku 4.5 via OpenRouter) assigns categories, confidence, and evidence for sampled WildChat turns.
3. **Disagreement mining:** High-confidence judge disagreements with the classifier are mined to correct false positives and category mismatches in subsequent training iterations.

3.3 Modeling and inference

The default classifier is an embedding-based model: SentenceTransformer embeddings (MiniLM) with a balanced logistic regression head. This design trades marginal accuracy for high throughput and easy calibration. The classifier outputs a predicted category and confidence for each assistant turn. Inference is batched and resumable, enabling millions of turns to be processed in offline monitoring mode.

3.4 Reporting and analysis

Downstream analytics compute:

- prevalence by category and confidence distribution,
- turn-index escalation trends,
- benchmark-to-reality gaps (JS/KL divergence and rank correlation),
- judge-classifier agreement and reliability diagnostics, and
- topic clustering with per-topic flag rates and escalation slopes.

4 Implementation

DarkPatternMonitor is implemented as a set of composable Python modules under `src/` with CLI entrypoints defined in `pyproject.toml`. The core implementation decisions are aligned with scale, reproducibility, and low operational cost.

Data ingestion. `src/ingest_wildchat.py` streams WildChat via the HuggingFace `datasets` API, extracts assistant turns with metadata (conversation ID, model, turn index), and supports resumable streaming to JSONL with sampling controls. `src/run_darkbench.py` loads DarkBench prompts from HuggingFace and executes elicitation with OpenRouter, OpenAI, or Anthropic clients.

LLM judging. `src/judge_label.py` implements a rubric-based judge prompt with JSON outputs (category, confidence, explanation, evidence). It supports parallel calls, resuming, and a reliability test mode that measures self-consistency across repeated judgements. Demo modes are provided when API keys are absent.

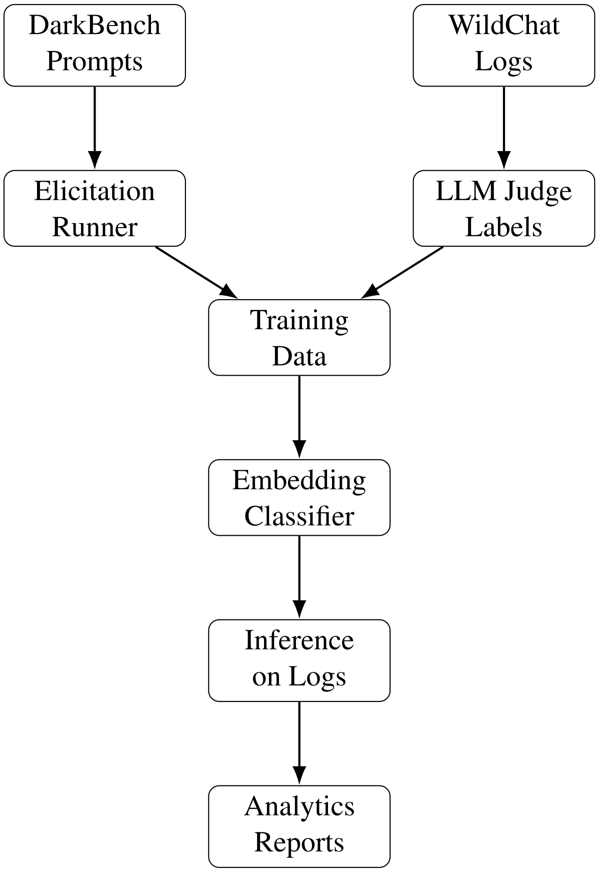

Figure 1: DarkPatternMonitor pipeline: benchmark elicitation and real-world logging feed labeling, training, inference, and analytics.

Training pipeline. The primary training path uses `src/train_classifier.py` with SentenceTransformer embeddings and a balanced logistic regression head. Optional transformer fine-tuning is available but secondary. Data preparation scripts (`prepare_training_data.py`, `prepare_v5_data.py`, `prepare_v5b_data.py`) combine benchmark labels, judge labels, and high-confidence corrections from classifier–judge disagreements.

Inference and analytics. `src/infer_wildchat.py` performs batched inference over JSONL turn files and writes predictions with class probabilities for later analysis. `src/analytics.py` computes prevalence and turn-index trends and emits figures. `src/gap_report.py` measures benchmark-vs-reality divergence. `src/reliability_report.py` reports judge–classifier agreement and failure modes. `src/topic_analysis.py` performs clustering-based topic analysis and per-topic escalation regression.

User interface. `app/streamlit_app.py` provides a lightweight dashboard for interactive inspection of detections and reports, enabling ad hoc scanning of transcripts and visualization of summary metrics.

5 Results

This section distinguishes **reported full-scale results** (from project documentation and changelog entries) from **repository artifacts** that are directly available in the current checkout. Where raw outputs are missing,

Table 1: Reported prevalence by category on 280,259 WildChat assistant turns.

Category	Count	Rate
Anthropomorphism	1,860	0.66%
Harmful generation	1,028	0.37%
Brand bias	405	0.14%
Sneaking	333	0.12%
User retention	53	0.02%
Sycophancy	47	0.02%
Total flagged	3,726	1.33%

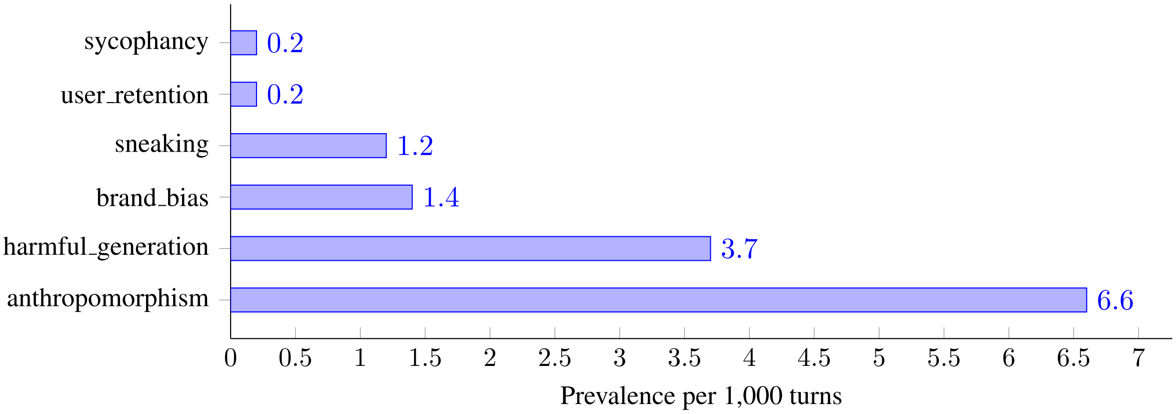

Figure 2: Reported per-1,000-turn prevalence by dark pattern category.

we mark reproduction as TBD.

5.1 Classifier performance

The final classifier reports validation accuracy of 78.7% with calibration error ECE=0.057 and high-confidence accuracy of 87.9%. The repository also includes an embedding-based classifier artifact (`models/classifier_full`) reporting macro F1=0.7802 on 326 evaluation samples. These figures are consistent with the stated design goal of prioritizing precision and calibration over aggressive recall.

5.2 Prevalence in WildChat

Project documentation reports a full-scale inference run over 280,259 assistant turns (100k conversations), yielding a 1.3% overall flag rate. Table 1 summarizes category counts and rates, and Figure 2 visualizes per-1,000-turn prevalence.

5.3 Model and turn-index effects

Results indicate higher flag rates for GPT-4 than GPT-3.5 (2.3% vs 1.6%), with a chi-squared test showing $p < 1e\text{-}36$. Turn-index analysis shows increasing flag rates across longer conversations (1.3% at turns 1–5 to 2.6% at turns 20+), and per-category escalation highlights sycophancy as the steepest riser (+42%). Figure 3 depicts the turn-index trend.

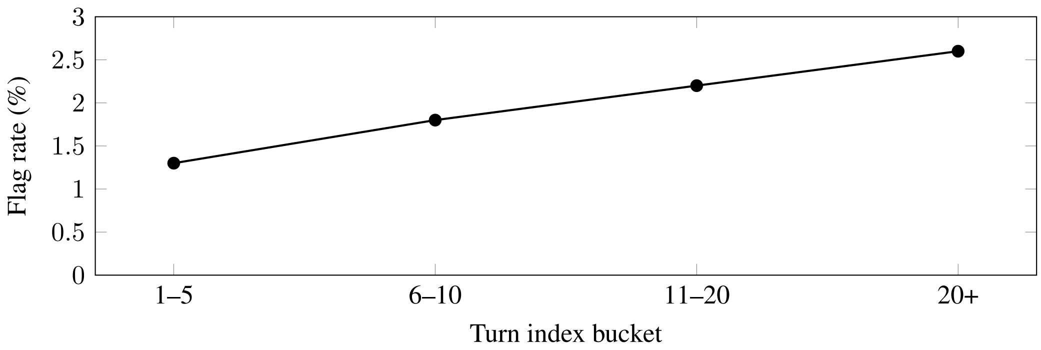

Figure 3: Escalation of dark pattern flag rates over turn index buckets.

5.4 Topic analysis

Topic clustering over 98,713 conversations identifies substantial heterogeneity. Character Interaction conversations show a 4.14% flag rate (roughly 5x the overall rate), while Coding Help is lowest at 0.08%. User Assistance shows the strongest escalation slope (p=1.62e-06). These findings motivate topic-aware monitoring thresholds.

5.5 Repository artifacts and reproduction status

The repository includes detection outputs and sample reports in `outputs/`. Full-scale outputs (280k detections) are available for download at the project website. The classifier, training data, and all pipeline scripts are included for full reproducibility.

6 Discussion

DarkPatternMonitor demonstrates that manipulation markers can be detected at scale with a practical trade-off between precision, throughput, and labeling cost. Results suggest that while the overall prevalence is low (1.3%), certain categories such as anthropomorphism and harmful generation appear consistently, and escalation patterns emerge in longer conversations. This supports a monitoring posture that is sensitive to topic and conversation length rather than uniform thresholds.

The benchmark-to-reality gap is a central insight: DarkBench elicits rare patterns at roughly uniform rates across categories, but WildChat shows substantial skew in real deployments. This mismatch implies that benchmark-only evaluation may miscalibrate operational risk. The project responds by calibrating with judge-labeled real logs and by mining disagreement cases to reduce false positives, aligning with the changelog evidence that precision-focused corrections significantly lower flag rates while improving calibration.

A notable systems implication is that the classifier is most useful as a triage layer, not a final arbiter of intent. The LLM judge provides richer explanations but is costlier and less consistent; the classifier is fast and consistent but context-limited. The hybrid approach can therefore be deployed as: (i) lightweight batch monitoring at scale, (ii) targeted judge review for high-confidence detections, and (iii) periodic retraining using disagreement mining to reduce drift.

All pipeline scripts, trained models, and full-scale outputs are available in the repository and project website, enabling independent reproduction and extension of this work.

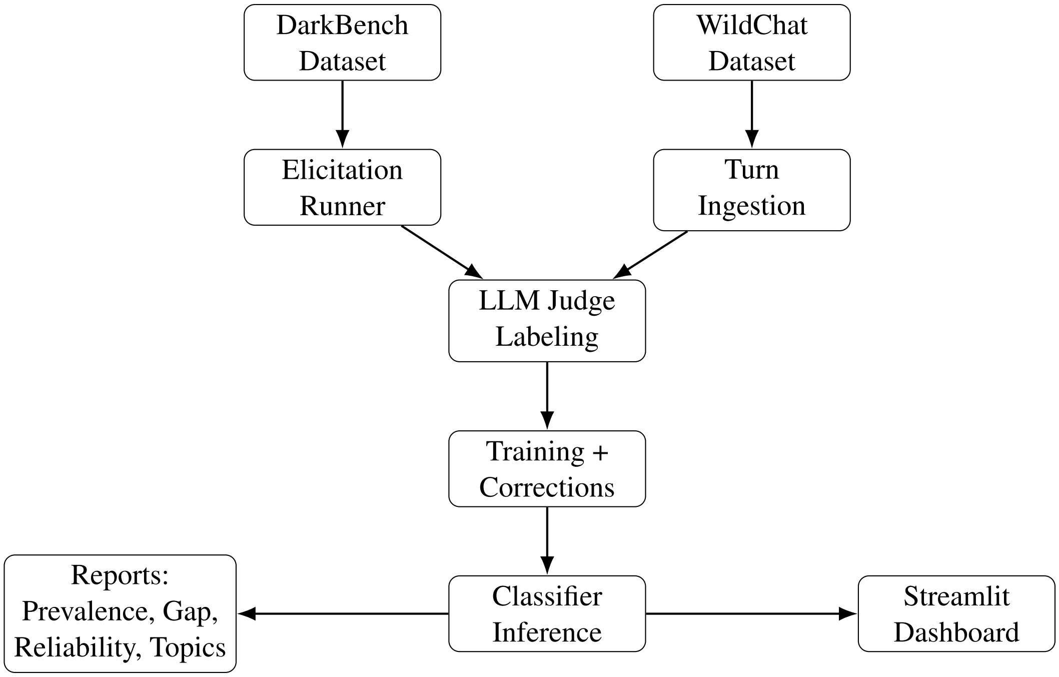

Figure 4: System architecture showing data sources, labeling, model training, inference, reporting, and the UI layer.

7 System Architecture

DarkPatternMonitor is organized as a modular pipeline with well-separated stages for data ingestion, labeling, modeling, inference, and reporting. Figure 4 depicts the end-to-end architecture used in the repository.

Key architectural properties include: (i) streaming ingestion with checkpointing for large datasets, (ii) parallel LLM calls for judge labeling and elicitation, (iii) a fast embedding classifier for batch inference, and (iv) analytics modules that reuse shared JSONL outputs. A lightweight Streamlit app provides interactive access to detections and reports without coupling to the core pipeline.

8 Conclusion

DarkPatternMonitor provides a practical, benchmark-calibrated pipeline for detecting manipulation markers in real LLM chat logs. By combining DarkBench elicitation, LLM-judge labeling, and a scalable embedding classifier, the system enables prevalence measurement, gap analysis, and reliability auditing at operational scales. Reported findings indicate measurable manipulation markers in production logs, increased rates in longer conversations, and strong topic-dependent variability.

Based on our findings, we propose a **three-tier monitoring framework** for production deployment: (1) *High-Alert* for character interaction and roleplay topics (5x baseline review rate), (2) *Escalation-Watch* for user assistance conversations exceeding 10 turns, and (3) *Standard* lightweight monitoring for technical and coding topics. This topic-aware approach enables efficient resource allocation for trust and safety teams.

Call for frontier model data. Our methodology is model-agnostic, but current analysis is limited to GPT-3.5/GPT-4 conversations from 2023–2024. We urge AI labs and researchers to collaborate on creating a WildChat-equivalent dataset for frontier models (Claude, Gemini, o1) with opt-in consent and regular updates.

Such a resource would enable the AI safety community to track manipulation patterns as model capabilities evolve and alignment techniques improve.

9 Code Availability

All code referenced in this paper is contained in the DarkPatternMonitor repository. Core entrypoints are provided via `src/` modules (ingestion, elicitation, judging, training, inference, and analytics), with a Streamlit demo in `app/streamlit_app.py`. Dependencies are specified in `requirements.txt` and `pyproject.toml`. The README documents how to download pre-trained models and outputs when available; these artifacts are not bundled in this Overleaf package.

References

[1] Llm-as-judge reliability for classification tasks, 2024. URL https://arxiv.org/abs/2412.12509. arXiv:2412.12509.

[2] Wildchat: 1m real llm conversations, 2024. URL https://arxiv.org/abs/2405.01470. arXiv:2405.01470.

[3] Darkbench: A benchmark for dark pattern elicitation in llms, 2025. URL https://arxiv.org/abs/2503.10728. arXiv:2503.10728.