

Deep Confidence For AI Safety¹

Saran Subramanian

Aqumen.ai

With

Apart Research

Abstract

Can confidence scores identify False Negatives, the most dangerous safety failures where models comply with harmful requests they should refuse? We explore whether low confidence during compliance signals unsafe outputs that should be diverted to refusals. Testing Qwen3-0.6B with DeepConf adaptive sampling on ToxicChat (5,083 instances) and WildGuardMix (1,725 instances), we examine confidence patterns across four experimental runs that compare a simple heuristic against the WildGuard 7B classifier.

Our 2×2 confidence matrices reveal a critical limitation. False Negatives, which correspond to unsafe compliance, show confidence scores (0.588–0.617) that are nearly identical to True Negatives, which correspond to correct safe responses (0.541–0.607). This overlap shows that confidence alone cannot identify safety failures without also filtering legitimate safe outputs. In other words, a pure confidence threshold would block a large number of harmless responses while still letting some harmful ones through, so the preventive gating hypothesis does not hold for this setting.

However, confidence still turns out to be useful in a different way. We find that adaptive trace generation achieves 64% token savings while maintaining 91% sensitivity for catching toxic prompts. Comparing simple heuristics against the WildGuard 7B classifier reveals only 1.7% accuracy improvement on ToxicChat, even though WildGuard introduces a 17 minute classification overhead in our setup. On WildGuardMix, the same classifier yields a 14.8% improvement (56.3% vs 41.5%), which shows that dataset quality is crucial when we evaluate safety classifiers.

Keywords: AI safety, false negative detection, confidence thresholding, preventive safety

1. Introduction

In AI safety, False Negatives (FN) occur when models comply with harmful requests they should refuse. These failures are especially worrying because they allow unsafe content to reach end users. In contrast, False Positives, where models refuse benign content, mainly frustrate or slow users down. As we move toward more capable models and higher stakes applications, it becomes increasingly important to understand whether we can detect the most dangerous failures in advance.

A natural idea is to look at the model’s own uncertainty. If a model is not very confident when it complies with a harmful request, perhaps we could use that low confidence as a warning signal. A Confidence-Based

¹Defensive Acceleration Hackathon Report

Safety Gate would then step in and force a refusal whenever the model’s confidence in its compliant answer drops below a chosen threshold.

DeepConf was originally developed and evaluated on AIME style math reasoning benchmarks, where confidence based truncation delivered substantial token savings without degrading solution accuracy. In this report we ask whether a similar strategy can transfer to a very different setting, namely safety classification. We use the DeepConf adaptive sampling framework to study confidence patterns for Qwen3-0.6B across two datasets, ToxicChat and WildGuardMix, and two refusal detection methods, a simple heuristic matcher and the WildGuard 7B classifier. We start with the hypothesis that False Negatives should exhibit lower confidence than correct outputs, including both True Positives (correct refusals) and True Negatives (correct safe compliance).

Our contribution is twofold. First, we provide a systematic empirical test of the preventive gating hypothesis and show that it fails in this regime. Our analysis of 2×2 confidence matrices shows that False Negatives and True Negatives share almost indistinguishable confidence distributions, so a simple threshold cannot separate unsafe from safe compliance. Second, we show that while confidence does not work as a gatekeeper, it does work as an efficiency signal. Adaptive sampling over ensembles delivers 64% token savings while keeping sensitivity for toxic prompts in the 91–92% range.

2. Methods

Experimental setup

We evaluated the Qwen3-0.6B model (0.6B parameters, open source) using the DeepConf adaptive sampling framework.

Datasets

- **ToxicChat** (LMSYS): 5,083 instances of real world user model interactions with human annotated toxic and safe labels, including a mixture of natural and mildly adversarial prompts.
- **WildGuardMix**: 1,725 instances with gold standard refusal labels on model outputs, constructed as a more explicitly adversarial stress test for safety classification.

DeepConf framework

For each input instance, DeepConf generates between three and ten traces using temperature sampling. From each trace, we extract per token log probabilities and compute a scalar trace confidence $\exp(\text{mean}(\log_probs))$.

We maintain an online percentile threshold over these confidence values. During evaluation we swept thresholds between the 20th and 90th percentile. For a given threshold, we generate traces one by one and stop early if the aggregated ensemble confidence exceeds the threshold, which induces token savings. Final per instance classifications are obtained by majority voting over the available traces.

Refusal detection methods

We compared two ways of mapping model outputs to refusal or compliance labels.

1. **Heuristic**: a lightweight pattern based matcher that searches for refusal phrases such as "I cannot" or "I am unable". This method is computationally cheap and easy to deploy.
2. **WildGuard**: a 7B parameter Mistral based classifier trained specifically for safety assessment, which we apply to the model outputs to obtain refusal decisions.

Experimental runs

We combined datasets and refusal detection methods into four runs.

- **Run 1:** ToxicChat + Heuristic (baseline)
- **Run 2:** WildGuardMix + Heuristic (adversarial test)
- **Run 3:** ToxicChat + WildGuard (classifier comparison)
- **Run 4:** WildGuardMix + WildGuard (gold standard evaluation)

Runs 1 and 2 apply the heuristic detector directly to the traces generated by Qwen3-0.6B. Runs 3 and 4 reuse the same underlying model outputs and apply the WildGuard classifier post hoc, so any differences between the heuristic and WildGuard reflect only the choice of classifier rather than resampling from the base model.

Metrics

Our primary metrics focused on safety and efficiency:

- **Sensitivity:** the proportion of toxic prompts that are correctly refused.
- **Specificity:** the proportion of safe prompts that are correctly allowed.
- **Token savings:** the reduction in inference cost achieved through adaptive sampling.

3. Results

3.1 Confidence patterns and the False Negative problem

Our first question is whether confidence scores can support a useful safety threshold. Across all four runs, we observed a consistent pattern in the confidence scores for different prediction categories.

True Positives, which correspond to correct refusals, consistently showed the highest confidence (0.650–0.682). This suggests that when the model correctly recognizes and refuses a harmful request, it does so with a strong internal signal.

The crucial problem appears when we compare False Negatives and True Negatives. In every experimental run, the confidence scores for False Negatives (unsafe compliance) were almost the same as those for True Negatives (safe compliance). In Run 4, which we treat as our most rigorous setting, the average confidence for False Negatives was 0.607 and the average confidence for True Negatives was also 0.607.

Because of this overlap, there is no helpful confidence threshold that catches False Negatives without also blocking a large fraction of safe content. Any threshold that is high enough to capture unsafe compliance will remove a similar proportion of safe, legitimate responses. This directly contradicts the idea that a model’s own confidence can serve as a simple proactive gate for the most serious safety failures in this regime.

Table 1: Run 4 (WildGuardMix + WildGuard) confidence matrix.

	Refusal (Pred)	Compliance (Pred)
Toxic (Actual)	TP: 0.663 (n = 515)	FN: 0.607 (n = 239)
Safe (Actual)	FP: 0.675 (n = 555)	TN: 0.607 (n = 416)

3.2 Efficiency gains through adaptive sampling

Although confidence did not work as a safety gate, it still provided useful information about how much computation we needed to spend. By using confidence to decide when to stop generating new traces, we achieved substantial computational savings without sacrificing safety.

Using a 20th percentile threshold, we reduced the average number of traces from ten to about 3.6 per instance. This corresponds to a 64% saving in tokens. The DeepConf framework allows filtering traces during generation (intra trace) or after generation (inter trace). We focused on an online adaptive sampling strategy, where we generate traces one at a time and stop once the ensemble’s aggregated confidence passes a threshold.

Most importantly, these savings did not degrade safety performance. Sensitivity remained in the 91–92% range across all runs. This shows that ensemble based safety systems can be made significantly more efficient in practice while still maintaining strong performance on toxic prompt detection.

3.3 Classifier performance: heuristic vs WildGuard

Finally, we compared the simple heuristic against the WildGuard classifier. On ToxicChat, WildGuard improved accuracy by only 1.7 percentage points over the heuristic, despite introducing a 17 minute classification overhead in our environment.

On the more adversarial WildGuardMix dataset, the story changed. Here, WildGuard improved accuracy by 14.8 percentage points, from 41.5% to 56.3%. This contrast highlights how important dataset choice is when we evaluate safety systems. Simple heuristics can look competitive on easier or more naturalistic data, yet they remain brittle under jailbreaks and adversarial prompts. Trained classifiers like WildGuard show their value more clearly when evaluated on datasets that are constructed to stress test refusal behavior.

4. Discussion

4.1 The limits of confidence based safety

Our results give clear evidence that raw confidence scores are not enough on their own for False Negative prevention in this setting. The overlap between uncertain compliance with harm and uncertain compliance with safe queries is too strong. This suggests that future work should explore alternative uncertainty measures, such as semantic entropy or disagreement between different models or prompts, instead of relying only on maximum probability.

4.2 The case for efficient ensembles

Even though confidence does not solve the gating problem, it does point toward a practical path for efficient safety. The 64% token savings we observe show that defensive ensembles do not have to be prohibitively expensive. By dynamically allocating compute, and spending additional traces only when the model is uncertain, we can keep most of the robustness benefits of ensembles at a fraction of the cost.

4.3 Recommendations

Taken together, these results suggest several design principles for confidence aware safety pipelines.

First, pure confidence thresholding on a single model should not be treated as a primary mechanism for preventing False Negative safety failures in this capability regime. The overlap between unsafe and safe compliance implies that any threshold with non trivial recall for harmful outputs will induce large and poorly calibrated false positive rates on benign content.

Second, confidence is more promising as a control signal for computational allocation than as a direct proxy for safety. Adaptive ensemble strategies that vary the number of traces as a function of confidence can recover most of the robustness gains of ensembles at significantly reduced token cost.

Third, hybrid architectures that combine inexpensive heuristics with more expressive but slower classifiers appear to be a reasonable compromise in practice. Our experiments suggest that simple pattern based detectors can handle many naturalistic cases, while higher capacity classifiers add value mainly on adversarial or carefully curated safety benchmarks.

These recommendations are not intended as prescriptive rules, but as empirically grounded guidelines for future work on Deep Confidence style methods in safety critical deployments.

5. References

- Fu, Y., Wang, X., Tian, Y., & Zhao, J. (2025). Deep Think with Confidence. *arXiv preprint arXiv:2508.15260*.
- Lin, Z., Wang, Z., Tong, Y., Wang, Y., Guo, Y., Wang, Y., & Shang, J. (2023). ToxicChat: Unveiling Hidden Challenges of Toxicity Detection in Real-World User-AI Conversation. *arXiv preprint arXiv:2310.17389*.
- Han, S., Rao, K., Ettinger, A., Jiang, L., Lin, B. Y., Lambert, N., Choi, Y., & Dziri, N. (2024). WildGuard: Open One-Stop Moderation Tools for Safety Risks, Jailbreaks, and Refusals of LLMs. *arXiv preprint arXiv:2406.18495*.

6. Appendix A: Security Considerations

Experimental Design Limitations

Label Mismatch: We compared input toxicity labels to output refusal behavior. This is a fundamental challenge in safety evaluation. A toxic prompt (e.g., "How do I stay safe online?") might receive a helpful, safe response. Our metrics count this as a False Negative (compliance with toxic input), even though the behavior is desirable. Future work requires datasets with gold-standard labels for generated outputs.

Assumption of Refusal: We assumed all toxic inputs should be refused. However, many "toxic" queries have legitimate educational or safety-seeking intents. Our current binary classification does not account for this nuance, potentially overstating the failure rate.

Technical Limitations

Model Scale: Our findings are based on Qwen3-0.6B. Confidence calibration and safety behaviors may differ significantly in larger models (8B, 70B).

Trace-Level Aggregation: We averaged confidence across entire responses. This masks token-level uncertainty spikes that might occur during specific harmful segments of a response. Per-token monitoring could yield better detection signals.